DOCUMENT RESUME

ED 409 790 HE 030 300

AUTHOR	Adelman, Clifford
TITLE	Leading, Concurrent, or Lagging? The Knowledge Content of Computer Science in Higher Education and the Labor Market.
INSTITUTION	National Inst. on Postsecondary Education, Libraries, and Lifelong Learning (ED/OERI), Washington, DC.; National Inst. for Science Education, Madison, WI.
SPONS AGENCY	National Science Foundation, Arlington, VA.
REPORT NO	PLLI-97-8046
PUB DATE	May 97
NOTE	68p.
CONTRACT	RED-9452971; 9453866
PUB TYPE	Reports - Descriptive (141)
EDRS PRICE	MF01/PC03 Plus Postage.
DESCRIPTORS	Associate Degrees; Bachelors Degrees; *Career Education; Cognitive Processes; Competency Based Education; Computer Literacy; *Computer Science Education; Core Curriculum; Curriculum Evaluation; *Education Work Relationship; Employment Opportunities; Employment Patterns; Graduate Study; Higher Education; Information Science Education; Instructional Effectiveness; *Labor Market; Labor Needs; Outcome Based Education; *Outcomes of Education; Program Effectiveness; *Relevance (Education); Undergraduate Study; Vocational Followup
IDENTIFIERS	Graduate Record Examinations; Knowledge Bases; Knowledge Development

ABSTRACT

This monograph traces changes in computer science as a field of study in the college curriculum, and the associated changes in the work force and labor market. The study followed two cohorts of students from high school to age 30. Records for the first cover the period 1972-84; and for the second, the period 1982-93. The purpose of the study was to: (1) access what knowledge students who earned Bachelor's or associate degrees in computer science took with them into the labor market, and (2) how well that knowledge met the needs of the labor market. Data used included: student undergraduate records, national examinations, curriculum statements of professional organizations, and graduate program offerings. Among findings was an extraordinarily high consensus between taxonomies of computer science courses and those of disciplinary organizations. The report discusses computer science course taxonomies, institutional providers, workplace requirements and high school experiences, number of students earning computer science degrees, other disciplines offering computer science-related courses, amount of time spent in related courses, data from Graduate Record Examinations, and career paths of graduates in computer-related fields. Twenty data tables are included and 11 notes provide additional information; an appendix contains technical notes on methodology. (Contains 64 references.) (CH)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

HE

ED 409 790

The Knowledge Content of Computer Science in Higher Education and the Labor Market

HE 030 300

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

This document has been reproduced as received from the person or organization originating it.

Minor changes have been made to improve reproduction quality.

Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

BEST COPY AVAILABLE

U.S. Department of Education
and
The National Institute for Science Education

Leading, Concurrent, or Lagging?

The Knowledge Content of Computer Science in Higher Education and the Labor Market

Clifford Adelman
Senior Research Analyst
Office of Educational Research and Improvement
U.S. Department of Education

U.S. Department of Education
Richard Riley
Secretary

Office of Educational Research and Improvement
Ramon C. Cortines
Acting Assistant Secretary

National Institute on Postsecondary Education, Libraries, and Lifelong Learning
Carole Lacampagne
Director

National Institute for Science Education, The University of Wisconsin-Madison, Funded by The National Science Foundation
Andrew Porter and Terrence Millar
Co-Directors

May 1997

The research reported in this monograph was supported, in part, by an interagency agreement between the National Science Foundation and the United States Department of Education (No. 9453866), and by a cooperative agreement between the National Science Foundation and the University of Wisconsin-Madison (Cooperative Agreement No. RED-9452971). At the University of Wisconsin-Madison, the National Institute for Science Education is housed in the Wisconsin Center for Education Research and is a collaborative effort of the College of Agricultural and Life Sciences, the School of Education, the College of Engineering, and the College of Letters and Science. The collaborative effort is also joined by the National Center for Improving Science Education, Washington, DC. Any opinions, findings, or conclusions are those of the author and do not necessarily reflect the views of the supporting agencies.

Contents

Acknowledgments	vii
Executive Summary	ix
Part 1. What Knowledge Is in Your Valise?	1
Part 2. The Story Told by Student Records	4
Part 3. The Story Told by Examinations	21
Part 4. The Story Told by Cohort Labor Market Experience	27
Part 5. Missing Issues in the Tracking Story	45
Part 6. Leading, Lagging, or Concurrent: A Summary	46
Notes	48
References	51
Appendix: Technical Notes and Guidance	57

Acknowledgments

This document is the second to evolve from a cooperative relationship between the Office of Educational Research and Improvement (OERI) and the National Science Foundation (NSF). Without the support and encouragement of NSF, particularly from Larry Suter and Mary Golladay, the principal data set on which the analysis is based would never be in the superb form it is in today. The relationship with NSF also lead to active participation in the life and research of the "College Level I" team of the National Institute for Science Education, in the context of which I have learned an enormous amount about how college faculty in the sciences, mathematics, and engineering are thinking about their teaching and student learning in the 1990s. Particular thanks go to Professor Arthur Ellis of the Department of Chemistry at the University of Wisconsin-Madison for facilitating that participation, as well as reviews of this document. These relationships demonstrate how federal agencies and the research centers they sponsor can work together to address problems we all care about.

This monograph was developed over a period of a year, in three distinct formulations. The first version was presented as a paper to the 1996 Forum of the European Association for Institutional Research, had a more comparative perspective and a greater emphasis on (higher education) institutional management. This version was also used as the reference point in the design of a NATO-funded cooperative research project with the Center for Science Research and Statistics (CSRS) of the Russian State Committee on Science and Technological Policy. In this form and these contexts, I received helpful feedback and suggestions from David Robertson of Liverpool John Moores University (UK), Joop van Schnijdel of the Hogeschool van Amsterdam, Dmitry Kostomarov of Moscow State University, and Levan Mindeli, Natalia Kovaleva, Leonid Gohkberg, Alexander Sokolov, and Natalia Gorodnikova—all of CSRS. The second set of reviews helped me recast the monograph for a domestic audience, and these—from Jerilee Grandy of the Educational Testing Service, Mary Vernon of the Department of Computer Science at the University of Wisconsin-Madison, and Charlotte Kuh of the Office of Science and Engineering Personnel of the National Research Council—provided new perspectives for the story lines and productive leads to the literature. Finally, a number of people gave generously of their time in reviewing the final draft, and helped me identify the chaff and clarify the presentation. They included Steve Ehrmann of the American Association for Higher Education, Greg Moses, Associate Dean of the School of Engineering at the University of Wisconsin-Madison, Anselm Blumer of the Department of Computer Science at Tufts University, and my colleagues Harold Himmelfarb, Nevzer Stacey, and Carole Lacampagne.

Lastly, all of us working in the vineyards of higher education owe much to the National Center for Education Statistics (NCES), which had the wisdom to include national college transcript samples in its longitudinal studies, thus providing a resource of knowledge that is simply unique in this world. With NCES come not only the data, but also my extraordinary colleagues and teachers, Nabeel Alsalam and C. Dennis Carroll, without whom our national knowledge would be so much less.

Executive Summary

This monograph traces change in the nature and extent of college students' study of computer science over the period, 1972–1993, the occupational destinations of students with computer science backgrounds, and the forces that shape the path from higher education to the labor market. Its fundamental question is whether what we teach is leading, concurrent, or lagging the state of the labor market in a field in which nothing sits still long enough to measure.

Using the college transcript samples from national longitudinal studies of two cohorts that were followed from high school to age 30, and curriculum statements and surveys of professional and disciplinary organizations, the study demonstrates that:

- The degree of consensus between an empirical taxonomy of computer science courses based on student records and rational taxonomies developed by disciplinary organizations is extraordinarily high. Over the past two decades we observe expansion of advanced topics, a growing emphasis on theory, and differentiation of subfields, all of which are indicators of a dynamic, maturing discipline.

- The increase in the proportion of undergraduates majoring in computer science (which hit a peak in 1986) accounts for only part of the increase in course taking in the field. A significant percentage of bachelor's degree recipients in other majors now complete mid-level courses in computer science.

- During the rapid growth of the period covered by the college student records, computer science determined its curriculum "gateways" in the categories of machine language and computer organization/architecture. The emergence of these gateways was equally clear at both the bachelor's and associate's degree levels.

- The "empirical core curriculum" of bachelor's degree recipients in computer science evidences continual change in the influence of engineering, mathematics, and business.

- Changes in the "empirical core curriculum" of associate's degree recipients in computer science evidence a considerable shift toward programming and applications. The mathematics background of these community college students, in both high school and college, improved dramatically in the 1980s.

Using surveys of graduate degree programs and studies of the Graduate Record Examination (GRE) field test in computer science, the monograph points out that

- The past two decades have witnessed both contraction in the ratio of subfields to programs at the doctoral level (an indicator of consolidation in a period of growth), and the emergence of a clear distinction between the scholarly (graduate level) and pedagogical (undergraduate) canons.

- The proportion of graduate programs (both master's and doctoral) requiring the GRE field test as part of the admissions process dropped considerably, a trend that corresponds to faculty judgment of a mismatch between the content of the examination and the existing undergraduate curriculum.

- Twenty-two percent of bachelor's degree recipients in computer science enrolled in formal graduate programs by age 30, principally in business and computer science, a ratio considerably below the average for bachelor's degree recipients in all fields.

The analysis of the labor market in light of student and curriculum history works in two directions: from the universe of students with backgrounds in computer science into the labor market, and from the universe of workers in computer-related occupations back into higher education. While these universes overlap, they are not identical. Using data from the National Science Foundation and the Bureau of Labor Statistics, as well as the longitudinal studies of the National Center for Education Statistics, the analyses point out that:

- The exponential growth in computer-related occupations has been uneven and complex, and, by the early 1990s, had not affected all industries.

- The early labor market experience of bachelor's degree recipients in computer science who continue into computer occupations is dominated by statistical reports, on-line computing, and development tasks. Their prior curricular experience has prepared them well for these tasks.

- The mix of computer-related occupations differs by industry, and, in judging potential dissonance in the labor market, the combination of occupation and industry is a critical gloss on students' courses of study. The publishing industry, for example, is a major employer of students trained in computer fields, but few of those students evidence any college-level study in journalism or graphic arts.

- Computer-related occupations are anomalies in terms of the formal educational attainment of workers, principally because the labor market in the field honors competence, content, and creativity more than credentials. For that reason, too, the "productivity orientation" of workers in computer-related occupations may be a more relevant indicator of potential than early career earnings and unemployment experience.

The monograph concludes that for those who concentrated in computer science and earned degrees at both bachelor's and associate's levels, the knowledge content of higher education was concurrent with the demands of the labor market, but that leading edges of the field are more likely to emerge outside formal education environments.

Part 1: What Knowledge is in Your Valise?

The most fundamental business of higher education is the dissemination of knowledge to students. Nearly two decades ago, Fritz Machlup broke new ground in our understanding of that "business" as an economic activity (Machlup, 1980). While he did not ask the questions directly, we do so today in his spirit: in relation to the needs of national and global economies, is the knowledge we impart lagging, concurrent, or leading? Can we even know? How? The questions—and their answers—are fundamental to the "market place" in which higher education exists. If the knowledge is lagging, economies stagnate and other providers of education may step forward; if it is at least concurrent, the signs of economic dissonance are few. Our students bring into the workplace what they have learned. As agents of supply, they can change the nature, processes, and substance of their work. For us to assist that change in more efficient ways, we need first to know precisely what is in the valises that they take with them, and how the contents of those containers are changing.

National education authorities in nearly all countries usually know no more than the program in which students are enrolled, for example, economics, electrical engineering, journalism (Organization for Economic Cooperation and Development [OECD], 1992). Where enrollments are not tallied by program, and in systems where students may change programs, authorities learn what students have studied only when credentials are awarded. Examples of both accounting systems (program enrollment and credential) are found in all the major national statistical reports on education (for example, Snyder, 1995; Gokhberg and Mindeli, 1995; Palafox, Mora, and Pérez, 1995). Neither system provides sufficient information for expert organizations, such as learned or professional societies, or for quality institutional management in higher education.

The issue is also one of mobility in a global economy. The scholarly disciplines are international information systems. Some of the learned societies in specific disciplines use surveys to describe the student experience of their fields. But these efforts are confined to disciplines with either strong knowledge paradigms (e.g., chemistry) or those with relatively few types of courses (e.g., "foreign" languages). In professional fields requiring licenses, we know the content of qualifying examinations, and national associations in those fields use this knowledge to determine whether an engineer, architect, or physician trained in one country may be eligible to practice in another (European Federation of National Associations of Engineers, 1975). Even when licensing is not a direct issue, some professions survey the characteristics of curricula, students, and degree requirements across national borders (e.g., Dorato and Abdallah, 1993). Again, though, these efforts are fragmentary, and far too occasional.

1.2 Problems and Purpose

This monograph traces change in one discipline, computer science, and the work force and labor market associated with that discipline. It asks what is in the valises that those who have studied in the field bring to the labor market and how that content is being regulated. It looks backwards from the labor market and asks where people working in computer-related occupations came from. That is its primary story. It also explores the virtues and

limitations of different types of unobtrusive evidence that are available to follow changes in the content of learning in *any* discipline, by offering the case of computer science as perceived through this evidence. Among the sources of educational information to be considered are:

- undergraduate student records,
- de facto national examinations,
- curriculum statements of professional organizations, and
- graduate program offerings.

Issues of data quality, depth, and reliability will receive attention as these data sources are utilized in the telling. Given a story line that starts in secondary school, expands in undergraduate study, and continues into work life, labor market data will also be used to explore how well we can track precisely where, in the economy, students take the knowledge they acquire in higher education. In the process, both usual and unusual suspects in the configuration of labor market "outcomes" will be considered, particularly something I call "productivity-orientation."

Along the way, the paper should force us to think about the following secondary questions:

- Given the fact that not all countries have the same type of sources of information (student records, syllabi, uniform examinations), can we extrapolate enough common descriptive elements of curriculum to determine whether the world's higher education enterprise is at least concurrent with the demands of a global economy?

- Many students leave higher education without credentials. They, too, enter the labor force with whatever knowledge they have gained in colleges, community colleges, and other institutions. How can we track the substance of their learning and factor it into our assessments of economic harmony?

- To what extent do cross-sectional labor market data and labor market histories drawn from longitudinal studies provide helpful information in determining whether what we teach in higher education is what the labor market needs?

1.2 Computer Life and Computer Work

However generic these questions, the story computer science and computer-related occupations drives them. Computer science was chosen as the case study field principally because it exhibits a dynamic degree of change over the past 25 years, change reflected in every one of the sources of information to be examined. It was also chosen because its history in relation to the labor market challenges conventional wisdom concerning the clean, vertical passage from college degrees in technical fields into the work force in the same fields. While the segments of this analysis dealing with higher education could be applied to other disciplines, for example, psychology, communications, and international studies, there

is no field in which the sources of information have been as challenging to stitch together in a coherent story governed by the phrase, "concurrent with the demands of a global economy." As astute observers of economic and technological life have observed, the computer has changed not only the infrastructure of work itself (Beniger, 1986; Barley, 1992; Cappelli et al, 1997), but also our entire culture (Pacey, 1983). Thus, those educated in the algorithmic bases, logic design, operating systems, and numerical analytic methods of the technology are at the core of this transformation: they are responsible for the hard and soft links in information technology environments, large and small.

Unlike professional fields such as medicine, nursing, or architecture, computer science presents distinct problems for labor market analysis (National Research Council, 1993). Not only are computing occupations changing, but so are job titles and occupational status (Braddock, 1995). Nothing sits still in an environment where software itself turns a professional software engineer into a mere technical applications analyst within two years. Ironically, computer design workers manufacture their own obsolescence—only to find themselves recalled to exalted positions when a problem from an older technology—such as programming mainframes to confront the year 2000—arises. What we sometimes interpret as gaps in the labor market may thus be cases of temporary recycling. And there is a great deal of computer-related work that falls under what Barley calls "the new crafts," that is, technical occupations in which degrees are incidental to formal education and in which "artisinal skills" are the key to productivity (Barley, 1992). Indeed, in many computer applications—and even systems development tasks—advanced formal education in universities is not required (National Research Council, 1993): the artisans of the computer trades can take a few courses and teach themselves the rest. There is a surfeit of autodidacticism in the computer work force.

Computer-related occupations can be—and are—subsumed under larger umbrellas of information technology. Other occupations, for example, those of business planning consultant-analyst, have also filled and churned with the content of information technology, so that their borders are always in flux. The librarian of 1980 who has learned search technologies and now prepares non-text sources (video, music) for classification may be called an "information archive systems developer." These titles are neither euphemisms nor inflations of reality: they describe what people do in an economy where information technology permeates the dailiness of worklife, and where computing applications themselves exist in a moving mixture of communications technologies.

Whether describing jobs or skills, the language of commentary on this labor market is filled with confessions of its own inadequacy. Job titles cover "heterogeneous activities" that are "difficult to describe and label" (National Research Council, 1993, p. 50). The "artisinal skills" of the new technological crafts "are difficult to verbalize, much less codify" (Barley, 1992, p. 15). It is important to note these difficulties at the outset because this monograph will not express frustration with change, labels, or the non-comparability of data. Our purpose is to explore the trail of the knowledge content of work as it flows from formal higher education in a specific technical discipline. At each stage of the pathway we will stop

and look around and describe the contours and texture of the path we have travelled. If we focus on contours and texture, we will be less bedeviled by terminology.

Part 2. The Story Told by Student Records

Student records hold significant information on the substance of courses of study. A national tapestry of course-taking data from two generations of students who participated in age cohort longitudinal studies is the source of this information. The first cohort was assembled when the students were in grade 12 in 1972. The second cohort was assembled when the students were in grade 10 in 1980, though this paper follows only those who reached grade 12 in 1982. The higher education records for the first of these cohorts cover the period, 1972–1984. The records for the second of these generations cover the period, 1982–1993. The basic data from this tapestry were published by the U.S. Department of Education in November of 1995 under the title, *The New College Course Map and Transcript Files* (Adelman, 1995). The data used here are slightly different because the editing of records for the 1982–1993 cohort was incomplete at the time of publication. For those unfamiliar with these cohorts, they are known as The National Longitudinal Study of the High School Class of 1972 (NLS–72) and the High School and Beyond/Sophomore Cohort (HS&B/So).

2.1 Taxonomies

The taxonomy of subjects in a data base of student records is derived empirically. That is, we examine the thousands of course titles used, for example, in computer science and computer-related fields, and sort those titles into bins. Each title is treated as an instance of course taking, and with each title comes information about the type of institution, the student's principal field of study, the credit value (or amount of time) spent in that particular course, the type and date of the term in which the course was taken, and a grade for student performance. The bins are then labelled, for example, Computer Networks, LANs [Local Area Networks], and given to a review group of faculty from the field of computer science, who offer suggestions ranging from relabelling to resorting to establishing new bins. In the case of computer science, it was determined that some bins belonged to engineering, business, and mathematics, and were passed to their respective disciplinary review groups. The data were then recoded and tested. With assistance from the National Science Foundation, this procedure was followed for 28 major fields of study in addition to computer science. Computer science was subject to two such examinations, in 1990 and again in 1995.

One of the ways to test the validity of the taxonomy is to match the final version against "competing" taxonomies used by professional or disciplinary organizations. Table 1 displays the "match" between *The New College Course Map* (CCM) categories and those used by Conference Board of the Mathematical Sciences (CBMS) in its 1990 survey of undergraduate computer science curricula in the United States (Albers, Loftsgaarden, Rung, and Watkins, 1992). The differences are minor. This degree of consensus between the empirical and rational taxonomies in the U.S. system of higher education is remarkable because there is no

Table 1.—Computer science course taxonomies: National Center for Education Statistics (NCES) and the Conference Board of Mathematical Sciences (CBMS)

CCM System (NCES), 1995[1]		CBMS 1990 SURVEY
110102	Computer Literacy, Computers and Society	Issues in Computer Science Computers and Society
110101	Introduction to Computer Sci	
110201	Computer Programming Programming in BASIC, PASCAL, FORTRAN, C, etc.	Computer Programming I Computer Programming II
110202	Algorithms, Computer Logic, Algebraic Language Programming	Algorithms
110203	Machine Language, Assembler Language, Computer Organization, Computer/Machine Architecture	Assembly Language Programming Introduction to Computer Organization
110204	Compiler Language, Grammar, Program Language Theory, Formal Languages	Survey of Programming Languages Compiler Design Formal Languages
110301	Data Processing, File Processing	Introduction to File Processing
110302	Data Structures, Discrete Structures	Advanced Programming and Data Structures Discrete Structures
110305	Computer Networks, LANs, Data Communication	Computer Networks
110401	Information Science/Systems/ Networks/Structure	
110402	Data Base Systems/Management	Database Management Systems

Table 1 (continued)

CCM System (NCES), 1995		CBMS 1990 Survey
110501	Systems Analysis/Development/ Design, Operating Systems, Systems Architecture	Introduction to Computer Systems Operating Systems Operating Systems and Computer Architecture I and II
110502	Software Engineering/Design/ Development/Methods	Software Design and Development
110601	Computer Applications: General, Software Applications	Introduction to Software Packages
110602	Computer Applications: Science and Engineering	
110603	Computer Applications: Business	
110604	Computer Applications: Other Fields	
110701	Simulation, Modelling, Parallel Processing	Modeling and Simulation Parallel Architecture
110702	Analysis/Theory of Algorithms, Automata, Automata Theory, Theory of Computation	Automata Theory
110703	Artificial Intelligence, Computer Vision, Expert Systems,	Artificial Intelligence, Expert Systems, Neural Nets
110704	Computer Graphics, Graphics Design	Computer Graphics
110801	Numerical Methods, Numerical Analysis, Mathematical Programming, Linear Programming	Numerical Methods, Numerical Mathematical Analysis, Numerical Math: Linear Algebra
110901	Other Determinable Topics	Other Computer Science, Semantics and Verification, Computational Linguistics, Complexity, etc.
270202	Discrete Math, Computer Math	Discrete Mathematics

central ministry defining degree programs and because the universe of higher education institutions is so diverse. This phenomenon indicates that the disciplinary organizations and the system of graduate training are playing a strong role in shaping the delivery of knowledge in the undergraduate curriculum.

The empirically derived taxonomies in most fields did not change during the two decades covered by these records. But in computer science, they did change. Table 2 sets forth course categories in computer science (beyond the basics) for the two age cohorts. We see both expansion in terms of advanced topics (simulation, modeling, artificial intelligence, graphics design), a growing emphasis on theory that is typical of maturing disciplines (Pantin, 1968), differentiation (information systems v. database systems), and refinement (digital logic included with algorithmic development and methods). All of these are indicators of a very dynamic field.

There is a difference, of course, between courses labelled in such a way that they are classified as computer science, no matter where they are taught within a college, and courses that include a considerable amount of computer skills (programming and applications) taught in other areas of the curriculum and labelled in such a way that they are *not* classified as "computer science." Good examples include Problem Solving in Chemical Engineering and Quantitative Research in Management. In both cases, one can reasonably assume that mastery of complex software packages, spread sheets, and graphics is either assumed or taught. But assumption is not certainty, and I would rather be sure. By limiting this analysis to computer science per se, we thus may under count the extent of student work in some fairly sophisticated applications courses.

2.2 Institutional Providers

When we turn to the question of where computer science topics are taught, the unit of analysis is the individual case of student enrollment in a specific topic in a specific type of institution. Whether the student masters the subject matter, (e.g. of computer organization or compiler languages), is irrelevant to this level of analysis. Whether the student even majors in the field, let alone completes a degree, are also irrelevant issues. The question is about the institutional agent of knowledge: *who* provides?

Why is this question important? In a hierarchical system of higher education, the concentration of provision at different levels of the system tells us what the discipline regards to be generalized or utilitarian as opposed to rare, advanced, and cutting-edge. In a field that matches different levels of skill demands in the labor market with educational programs, a standardized structure of provision is established. We know whether the provision is efficient, whether education in the field is leading, concurrent, or lagging, when the leading edge is found only at the highest level and when what is most common is found at the most common level.

Table 2.—The changing shape of computer science as an undergraduate discipline, 1972–1993

NCES COURSE CATEGORIES: 1972–1984	NCES COURSE CATEGORIES: 1982–1993
Algorithms, Algorithmic Methods, Algorithmic Development	SAME plus Computer/Digital Logic
Machine Language, Assembler Language	SAME plus Computer Organization, Computer/Machine Architecture
Compiler Language, Grammar, Program Language Theory	SAME plus Formal Languages
Data/File Processing	SAME
Data Structures	SAME plus Discrete Structures
NO CATEGORY	Computer Networks, Data Communication
Information Science/Systems, Data Base Concepts/Management	SAME minus Data Base
NO CATEGORY	Data Base Systems/Management
Systems Analysis/Development, Computer Networks/Communication	SAME minus Computer Networks, plus Operating Systems, Systems Architecture
Systems Software/Programming, Software Methods/Design	Software Engineering/Design, Software Methods
Advanced Topics (AI, Graphics, etc.)	NO CATEGORY
NO CATEGORY	Simulation, Modelling
NO CATEGORY	Theory of Algorithms/Automata/Computation
NO CATEGORY	AI, Expert Systems, Computer Vision
NO CATEGORY	Graphics Design
Numerical Methods/Analysis, Linear Programming, Mathematical Programming	SAME
Other Determinable Topics	SAME

SOURCES: *A College Course Map* (Washington, DC: U.S. Department of Education, 1990) and *The New College Course Map* (Washington, DC: U.S. Department of Education, 1995).

Table 3 provides illustration of this standardized structure of provision. The first row, for "All Courses," is the benchmark: of *all* cases of course taking in *all* subjects, this is the distribution by institutional type. The data for any specific course must be judged against this empirical norm. We expect doctoral degree granting institutions to be the principal providers of course work in artificial intelligence, and with a 72 percent share of enrollments (measured against a 34 percent share of enrollments in all courses in all disciplines), we are not disappointed. At the same time, we do not expect two-year degree programs to offer upper-level courses such as simulation or expert systems. We do expect two-year colleges that prepare mid-level technical workers to provide the bulk of training in data processing. Again, we are not disappointed. In both generations of students, two-year colleges held a 56–57 percent share of data processing enrollments (against a 23 percent share for all courses).

Changes in the share of enrollments enjoyed by specific types of institutions from one student cohort to the next tell us something of changes in the discipline and its delivery system. In terms of general delivery, it is obvious that liberal arts colleges began to teach computer science in the 1980s, hence took a piece of the national share from all other types of institutions.

But enrollment share by course category is more revealing of the nature of provision, and, as table 3 shows, this shifted considerably in Introduction to Computer Science and in the category that covers Machine/Assembly Language and Computer Organization. Doctoral and Comprehensive universities "lost" share in the introductory course but gained share in what I will call the "gateway" courses. Precisely the opposite trend occurred in community colleges, a puzzling development in light of the increase in credits earned in both categories by associate's degree recipients (see the discussion of table 7 below).

How do we explain these changes? First, by 1982, when the second age cohort was beginning its higher education career, introduction to computer science and computer programming were part of the elective curriculum in many U.S. secondary schools. Nearly 16 percent of students in this age cohort who entered higher education had studied computer science before they arrived in college. Of that group, over 80 percent attended doctoral degree granting or comprehensive universities. They did not need the introductory course, therefore skipped over it and enrolled in mid-level courses, most particularly machine/assembly language and computer organization/computer architecture. At the same time, the computer science curriculum in community colleges was paying more attention to applications and less to theory.

Table 3.—Change in distribution of enrollments in selected computer science course categories, by institutional type: 1972–1993

	Doct.	Comp.	LibArt	CommC	Special	Trade
All Courses	**34%**	**32%**	**7%**	**23%**	**3%**	**1%**
Introduction to	32	38	5	17	7	Low N
Computer Science	**25**	**35**	**8**	**27**	**3**	**2**
Computer Programming:	25	32	3	33	4	3
All Languages	**22**	**33**	**6**	**31**	**6**	**2**
Machine/Assembly	25	23	2	37	2	11
Language; Computer	**31**	**42**	**4**	**18**	**4**	**Low N**
Organization/Architect.						
Data Processing	11	22	2	56	2	6
	11	**20**	**3**	**57**	**4**	**5**
Data/Discrete	44	46	Low N	7	4	0
Structures	**42**	**47**	**4**	**4**	**3**	**0**
Simulation,	---	---	---	---	---	---
Modeling	**50**	**39**	**7**	**0**	**2**	**2**
Artificial	---	---	---	---	---	---
Intelligence	**72**	**23**	**3**	**Low N**	**Low N**	**0**

Institutional Type:
Doct = Doctoral. Ph.D. granting.
Comp = Comprehensive. Grants up to Master's degree; covers both traditional arts and sciences and occupational programs.
LibArt = Liberal Arts. Bachelor's degree only.
CommC = All 2-year, associate's degree granting schools.
Special = Specialized, e.g. degree granting schools of art, music, mining, fashion.
Trade = Non degree granting vocational schools.

NOTES: (1) All rows add to 100 percent.
(2) Figures in **bold** are for the 1982–1993 cohort.
(3) Figures in light type are for the 1972–1984 cohort.
(4) "Low N" indicates cells in which the unweighted number of cases was too low to produce an estimate.

SOURCES: *A College Course Map* (Washington, DC: U.S. Department of Education, 1990) and *The New College Course Map* (Washington, DC: U.S. Department of Education, 1995)

2.3 The Force of Workplace Requirements and High School Experiences

A second explanation of these changes derives from the fact that direct instruction in computer science or programming was not the only influence on student course-taking behavior. By the early 1980s, basic computer literacy was an increasing demand in the workplace and computer skills were required in other disciplines as well. By 1984, among the cohort who graduated from secondary school in 1982 and who entered higher education:

- one-third had used a mini-computer or mainframe computer for either education or job related activities;

- 43.5 percent had programming experience, and 19.8 percent in languages other than BASIC; and

- 7.3 percent had used what, at the time, were considered complex statistical software packages.

This information comes not from student records, rather from surveys that were part of the High School and Beyond longitudinal study. While it is not clear that students knew the difference between a micro-computer and a mini-computer (two of the hardware categories used on the survey questionnaire), they certainly knew the difference between COBOL, PASCAL, and BASIC. Comparable data for the cohort that graduated from secondary school a decade earlier are obviously not available. We never thought to ask such questions in the 1970s because, at the time, the realities to which they refer were the exclusive properties of research labs, the military, computer manufacturers, and data processing firms.

Not surprisingly, the high school mathematics backgrounds of students with computer experience by age 20 (in 1984) and who had continued their education after high school indicate a strong quantitative bent. For the group that had used a mini-computer or mainframe, 31 percent had studied math at the level of trigonometry or higher (precalculus, calculus) in high school versus 18 percent among those with no computer experience. For those with programming experience in languages other than basic, the proportion reaching this high level of secondary school mathematics was 44 percent (versus 13 percent of those with no programming experience whatsoever).[2] While the mathematical foundations of computer science are found more in finite and discrete mathematics than calculus, the training in logic and quantitative reasoning afforded by advanced study of mathematics in high school well-serves—and even prods—future study and further work in computer-related fields.

2.4 Participation in Learning: Volume

Using student records, there are two ways to analyze the extent and depth of learning in specific fields and subfields. The first describes participation: the proportion of any group of students who successfully completed a course in X. The second describes depth and

concentration: of a given group of students, how much of their total time in higher education (using credits as a proxy measure for time) was spent studying X? In using records as archival indicators of change, the second question is more important because time on subject matter is a direct indicator of depth of study, and, because undergraduate time is finite, the comparative share of time in various courses tells us what topics are being emphasized.

The umbrella for consideration of changes in rates of participation and depth involves degrees conferred. In order to judge the diffusion of knowledge in a field, we need to compare the amount of study by concentrators in that field to that of students who concentrated in other fields. Table 4 sets out the basic changes between the two cohorts in the proportion of bachelor's degree completers who concentrated in computer science and related fields. The differences are very dramatic: for computer science alone, from 0.6 percent of all bachelor's degrees awarded to 4.2 percent; for all computer-related fields, from 1.0 percent to 6.1 percent.[3]

Table 4.— Changes in proportion of bachelor's degree recipients earning degrees in computer-related fields, by age 30.

	COHORT A: 1972–84			COHORT B: 1982–93		
Degree Field	**Men**	**Women**	**All**	**Men**	**Women**	**All**
Computer Science	0.9	0.2	0.6	5.6*	3.0*	4.2
Computer Programming	—	0.1	<0.1	0.2	0.1	0.1
Computer Engineering	0.3	0.0	0.2	1.1	0.1	0.6
Operations Research (Business)	0.2	0.1	0.1	1.5	0.7	1.1
Total	**1.4%**	**0.4%**	**1.0%**	**8.2%***	**4.0%***	**6.1%**

*Statistically significant difference. $p \leq .05$

SOURCES: National Center for Education Statistics, "National Longitudinal Study of High School Class of 1972," special analysis file. National Center for Education Statistics, "High School and Beyond/Sophomore Cohort," special analysis file.

The percentages may appear small, and one cannot claim statistical significance for any comparison of subfields with less than one percent shares of degrees, but when one translates these percentages into weighted numbers, the totals are not insignificant. Given the modal years of bachelor's degree completion for the two longitudinal studies cohorts of 1976 and 1986, the weighted totals for computer science degrees alone match cross-sectional degree data reported by institutions (Snyder, Hoffman, and Geddes, 1996) very closely:

Number of Bachelor's Degrees in Computer Science

	Computer Science Degrees Reflected in Longitudinal Studies	**Computer Science Degrees Reported by Institutions of Higher Education**
1976	5,383	5,652
1986	39,860	39,589

With this growth as a background, it is not surprising to find the following changes in participation rates (proportion of students successfully completing a course in . . .) for bachelor's degree recipients:

Undergraduate Course Participation Rates for All Bachelor's Degree Recipients (by Age 30) in Two Cohorts

	1972–84 Cohort	**1982–93 Cohort**
Introduction to Computer Science	8.4%	22.2%
Computer Programming	10.2	25.0
Algorithms, Computer Logic	0.8	2.6
Machine Language/Computer Organization	0.9	5.3
Applications: Science and Engineering	0.7	4.6
Applications: Business	1.4	6.5

These increases cannot be explained wholly by students who completed degrees in computer science. As table 5 demonstrates, business majors (26 percent of all bachelor's degree recipients for the 1982 cohort), engineering and architecture majors (9 percent), and majors in physical sciences (2.5 percent) were responsible for a significant proportion of the increase in participation rates. By the third course, the student is studying either applications or one of the middle-level "gateways" in the computer science field. These students, majoring in other fields, are filling their knowledge valises with computational protocols, hardware

specifications, logic designs, and techniques of data editing and imputation—all of which eventually enter the workplace, with the students, in a broad range of occupations and industries. Even among those in this cohort who earned bachelor's degrees and were working in computer-related occupations in 1991, 57 percent majored in fields *other than* computer science (10 percent, in fact, majored in humanities, arts, or social sciences).

Table 5.—Extent of college-level study in computer science by bachelor's degree recipients who majored in other fields, 1982–1993 cohort

	Number of Completed Courses in Computer Science					
Major Field	**None**	**1**	**2**	**3–4**	**5–6**	**7+**
Engineering/Architecture	12%	31%	21%	26%	7%	4%
Business	18	41	20	13	4	4
Physical Sciences	34	28	16	17	5	0
Life Sciences	55	31	9	4	Low N	0
Social Sciences	59	25	12	3	Low N	Low N
Applied Social Sciences	60	28	8	Low N	Low N	Low N
Humanities	67	18	10	Low N	Low N	Low N
Health Sci & Services	72	20	4	4	0	0
Arts	74	18	5	Low N	0	0
Education	74	21	2	Low N	0	Low N
All	**43**	**28**	**12**	**8**	**2**	**6**

NOTES: (1) "Low N" indicates cells in which the unweighted number of cases was too low to produce an estimate.
(2) Rows may not add to 100.0% because of the "Low N" cells.
(3) "Applied Social Sciences" includes communications, criminal justice, public administration, social work, and home economics.

SOURCE: National Center for Education Statistics, "High School & Beyond/Sophomore Cohort," special analysis file.

2.5 Discovering the Gateways

Instruction in computer science in the 1970s and early 1980s was influenced heavily by the engineering profession. This was natural, either because most upper-division courses in the field were housed in engineering schools or, where computer science was taught wholly outside mathematics departments, engineering schools provided the administrative umbrella. Given the requirements for program accreditation in engineering, model curricula often followed American Board of Engineering and Technology (ABET) accreditation guidelines (Tucker, et al, 1991).

Program accreditation is important in fields that require licenses to practice, and engineering is one of those fields. But, with some exceptions (for example, business administration), program accreditation is not eagerly sought by most colleges and universities for non-licensure fields. The reason is simple: the accreditation process is time consuming and expensive. If the result of the process does not measurably enhance the credentials awarded to students, institutions will not expend the effort. In the case of computer science, there was little incentive for colleges that did not offer engineering degrees to follow the guidelines for computer science programs (for example, those of 1983) of the Institute for Electrical and Electronic Engineers (IEEE), a member organization of ABET.

What did not happen by formal quality assurance mechanisms happened naturally in the field of computer science: the emergence of a configuration of topics as "gateways" to full undergraduate specialization in the field. The establishment of "gateways" in any field is an important step in the evolution of a pedagogical canon (Mueller, 1989). The "gateway" course or courses are middle-level, and focus on the major tools and theory needed by advanced students. The "gateways" often—but not always—serve as well to sort students. That is, they occur at a point in students' educational careers where decisions are made to specialize, and they are placed at a level of complexity and demand through which students learns whether they will succeed in the field.

Experimental Psychology and the statistics that accompany it, is a good example of a gateway in a mature field. No matter how fragmented the discipline of psychology sometimes appears, no matter how many competing paradigms one may find in the field, experimental methods and statistics are still at the core of the discipline. The same observation can be made about the position of microeconomic theory and macroeconomic theory in economics, or physical chemistry in chemistry. When you finish any of these courses, you know whether you want to concentrate in psychology, economics, or chemistry and (perhaps) how well you will perform for the remainder of your program. When and if the disciplines of psychology, economics, and chemistry change their knowledge paradigms in enduring ways, the gateway will change as well.

By the mid-1980s, a configuration of mid-level courses in computer science had emerged as gateways to full specialization. The evidence is empirical and unobtrusive: enrollment volume along the definition of "mid-level" used in the cross-sectional surveys of departments

conducted by the Conference Board of Mathematical Sciences. From this source, the courses (and their national enrollments) were:

Assembly Language Programming	24,000
Data Structures	24,000
Introduction to Computer Systems	18,000
Introduction to Computer Organization	14,000

(Albers, Anderson, & Loftsgaarden, 1987)

Enrollments, however, are not completions. Referring to the national transcript sample for the 1982 cohort, the mid-level courses that emerged as sorting mechanisms were those with the highest rates of non-completion by either withdrawal (Data Structures at an 11.5 percent withdrawal rate) or failure (Assembly Language Programming and Introduction to Computer Organization at an 11.4 percent failure rate). In the taxonomy described in section 2.1, these two courses are in the same category. The failure rate in that category, 11.4 percent, was the highest of *any* course category in the national taxonomy of 1,037 courses in all fields. Whether or not computer science faculty were fully aware of the effects, they were using these mid-level courses in a triage function.

By 1990, when overall enrollments in computer science had declined, Introduction to Computer Systems fell out of the gateway configuration. Computer Architecture (as taught in computer science departments, as opposed to engineering schools) appears to have replaced it (Albers, Loftsgaarden, Rung, and Watkins, 1992). The taxonomic categories used by the Conference Board of Mathematical Sciences are a bit fuzzy on this issue, but when we observe enrollment holding steady or rising in the Computer Architecture categories at the same time that enrollments are declining in the vast majority of other course categories, we can hypothesize that the pedagogical practices of the field were reflecting the growing consensus in professional communications concerning undergraduate curriculum (Mulder and Dalphin, 1984; Gibbs and Tucker, 1986; Denning et al, 1989).

2.6 Depth of Learning: Time

The emergence of the gateway courses is related to the second archival indicator of the precise content of student learning in a field: time-on-subject-matter. Using credits as a proxy measure, we take the total amount of time a student spends in higher education, and ask what proportion of that time was spent in a specific course. Tables 6 and 7 present this information for students who earned degrees at the bachelor's and associate's levels.

Table 6 displays the "top 25" course categories accounting for one percent or more of bachelor's degree recipients' undergraduate time in either the 1972–84 cohort or the 1982–1993 cohort, and arranged according to their rank in the elder cohort. For both age cohorts, the 25 categories account for a very high percentage of the students' total time (in the taxonomies used, there were over 1,000 course categories).[4] At the bachelor's degree

Table 6.—The empirical core curriculum of students completing bachelor's degrees in computer science.

	1972–84 Cohort	1982–93 Cohort	
Course Category	**% of Time**	**% of Time**	**Change**
Calculus	8.7%	6.7%	-2.0%
Computer Programming	4.4	5.2	+0.8
General Physics	3.4	2.3	-1.1
English Composition	3.3	3.1	-0.2
Computer Engineering	2.5	1.1	-1.4
Intro. Computer Science	2.4	2.3	-0.1
Numerical Methods	2.2	1.0	-1.2
Software Engineering	2.0	1.0	-1.0
Systems Design/Software	2.0	3.2	+1.2
Introduction to Economics	1.9	2.8	+0.9
Data Structures	1.9	2.0	+0.1
Statistics	1.8	1.7	-0.1
Computer Organization/Archit	1.6	2.6	+1.0
Post-Calculus Math	1.5	1.6	+0.1
Introduction to Engineering	1.5	0.5	-1.0
Electrical Engineering	1.4	0.5	-0.9
Compiler/Program Languages	1.5	1.3	-0.2
U.S. History Surveys	1.3	1.4	+0.1
General Chemistry	1.1	1.5	+0.4
Operations Research	1.1	0.4	-0.7
Data Base Systems	1.1	1.2	+0.1
Discrete Mathematics	1.0	1.1	+0.1
Introduction to Psychology	1.0	1.5	+0.5
Introduction to Accounting	1.0	1.7	+0.7
Precalculus	0.9	2.5	+1.6
Total % of All Credits	**52.5%**	**50.2%**	**-2.3%**

SOURCES: NCES, "National Longitudinal Study of the High School Class of 1972" and "High School and Beyond/Sophomore Cohort," special analysis files.

level, there is very little change in the percentage of total time claimed by some subjects from one cohort to the next. The difference between 1.9 percent and 2.0 percent in any course category is really no difference at all; whereas, when one is measuring proportions of total time, a change of 0.5 percent or more is substantial. The shorter the proxy time, however, as is the case of the associate's degree, the more volatile the changes.

What do we see? First, the apparent decline of the influence of engineering on the composition of the pool of people concentrating in computer science. For the 1972 cohort, 3 of the 25 top "time" courses were in engineering; for the 1982 cohort, only 1 remains, and that course category (computer engineering) dropped from 2.5 percent of total undergraduate time to 1.1 percent.

As the influence of engineering appeared to wane, however, the position of mathematics held steady, but fell in level. That is, for example, we see more time in precalculus (and, had the list been expanded to the "top 30" courses, we would see more time in college algebra as well) and less in calculus. As for the interaction between business administration and the computer science curriculum, the drop in Operations Research is offset, in part, by a rise in Computer Applications: Business (another course beyond the "top 25"). But this shift is similar to that in mathematics: it's a drop in the level of analytic quantitative demands, and perhaps a tribute to the increasing power of business related software. What happened? The mathematics requirements for business degrees rose during the 1980s to a minimum of college algebra (American Assembly of Collegiate Schools of Business, 1991), and the business and computer science curricula penetrated each other. Some students evidently started out in business, satisfied the mathematics requirements in business, and then switched over to computer science, where discrete mathematics is the underlying paradigm.

In describing these empirical trends in course work in terms of the comparative influence of engineering, business and mathematics, I qualified the observation with the verb, "appear," particularly as the Systems Design/Software category will include some software engineering (for a discussion of curricular interpenetration in this field, see Wulf, 1992). During the 1970s and 1980s, the discipline struggled out of its dual-origins in mathematical principles and the kinds of heuristics, design decisions, and applications that are at the core of engineering. The tensions evident in the literature and the exploration of boundaries and commonalities between computer science and computer engineering (e.g. Denning, 1985; Loui, 1987) are indicative of a maturing discipline discovering where it sits in academic organizations. What appears to be the case in student course taking may thus reflect transitional stages in the development of a pedagogical canon of a discipline seeking its paradigmatic balance of theory, design, analysis, and implementation.[5] From a more practical point of view, it may be simply a case of computer science programs established in the 1980s in colleges without engineering schools.

For students whose highest degree (by age 30) was the associate's (table 7), the changes from the 1972 to the 1982 cohort evidence a clear differentiation of program purpose toward programming and applications. Because there is a great deal more volatility in the "top 25"

Table 7.—The empirical core curriculum of students completing associate's degrees in computer fields.

	1972–84 Cohort	1982–93 Cohort	
Course Category	**% of Time**	**% of Time**	**Change**
Computer Programming	11.7%	18.0%	+6.3%
Introduction to Accounting	7.4	7.8	+0.4
Data Processing	6.6	7.1	+0.5
Systems Software	1.1	5.5	+4.4
English Composition	4.6	3.9	-0.7
Comput Organization/Archit	1.7	3.2	+1.5
Introduction to Psychology	2.1	2.9	+0.8
Introduction to Economics	2.6	2.2	-0.4
Introduction to Business Admin	1.2	1.9	+0.7
Computer Applics: Business	0.4	1.7	+1.3
Remedial English/Writing	0.9	1.7	+0.7
Computer Applics: General	0.6	1.5	+0.9
PreCalculus	0.4	1.4	+1.0
Calculus	1.1	1.4	+0.3
Introduction to Computer Science	1.1	1.3	+0.2
Introduction to Communications	1.4	1.2	-0.2
Advanced Accounting	1.8	1.2	-0.6
History: US or World	2.2	1.1	-1.1
Business Law	1.3	1.1	-0.2
Discrete Math	1.5	1.0	-0.5
PreCollege Algebra	2.7	0.9	-1.8
Statistics	2.1	0.9	-1.2
U.S. Government	1.2	0.8	-0.4
Data/Discrete Structures	0.3	0.8	+0.5
Business Math: College Level	0.2	0.8	+0.6
Total % of All Credits	**58.2%**	**71.3%**	**+13.1%**

SOURCES: NCES, "National Longitudinal Study of the High School Class of 1972" and "High School and Beyond/Sophomore Cohort," special analysis files.

courses at the associate's degree level, they have been arranged according to their rank in the experience of the more recent cohort (1982–1993). Comparing the two halves of the ledger, the number of computer science courses jumped from five to nine (for example, Computer Applications: Business, Computer Applications: General, and Data Structures accounted for comparatively little time in the curriculum of the 1972–1984 cohort). Overall, the time spent in utilitarian skills such as programming and applications rose from 11.7 percent to 21.2 percent. It is also obvious that "computer-related" at the level of the associate's degree in both cohorts meant "combined with business and accounting," as courses in those fields totalled 12 percent of student time.

In terms of backgrounds and achievement in mathematics and communications skills, we witness contrary trends. The quality of associate's degree recipients' secondary school mathematics training increased, as evident by the comparative shares of calculus and pre-calculus versus precollege algebra on the time continuum. However, the data document an unfortunate decline in the communication skills of associate's degree recipients in computer-related fields. The proportion of time spent in regular English composition dropped by the same amount as the proportion of time in remedial writing rose. With this exception, the empirical curriculum for associate's degree students in the 1980s appears to be concurrent with generalized skills required in mid-level computer support occupations. Indirect evidence from professional associations, though, indicates that by the early 1990s, the curriculum for specialists at the associate's level was lagging. In 1993, the Association for Computing Machinery (ACM) published guidelines for "knowledge units" for nine subject areas presented in two-year college computer science programs both for students seeking terminal associate's degrees and those preparing for transfer to four-year colleges (ACM, 1993).

The system of content analysis used by the ACM committee, following the model of its predecessors (Tucker, et al, 1991), provides an alternative to the method we used to sort student records in the national transcript samples. The alternative is conceptually attractive but difficult to actualize. In offering models of curricula, the ACM committee took each course, and presented the amount of time within the course to be devoted to each of the nine subject areas (where applicable). The committee then totalled that time across all courses.

Recommended Time in "Knowledge Units": Associate's Degree in Computer Science

Knowledge Unit	Percent of Time
Algorithms & Data Structures	27%
Architecture	15
Artificial Intelligence/Robotics	4
Database and Information Retrieval	2
Numerical and Symbolic Computation	3
Operating Systems	3
Programming Languages	22
Software Methods and Engineering	19
Social, Ethical and Personal Issues	5

(ACM, 1993, p. CS-71)

While this presentation is not wholly comparable to that of table 7, it does imply a greater emphasis on algorithms, data structures, and problems of space-time complexity and less emphasis on database and information retrieval than indicated in the empirical curriculum of associate's degree students in the 1982–1993 period. Were students to follow this model, they would have greater flexibility in adapting to new programming languages and computer environments than they would under the older applications paradigm evident in table 7.

2.7 Limitations of This Analysis

There are obvious limitations to an analysis of curricula based on the ostensible—and broad—topics of courses or even guidance of learned societies concerning curricular content. A course in systems design, for example, can walk students, in painstaking detail, through examples of past systems designs or present exercises in construction of systems designs with analyses of trade-offs at every branch of the process. Certainly, a course could do both, provided that students had first mastered systems concepts (Anderssen and Myburgh, 1992). It could even simulate expected changes in technology, for example, 600 megahertz mother boards for personal computers and work stations. The longitudinal studies on which we are relying for the bulk of this analysis are constructed as relational data bases. A course with the topic, "relational data bases," can be a highly applied affair or a more theoretical exploration of such notions as "nesting" and hierarchy of files. Our sources simply don't know what is being played out under these roofs and how, hence it is difficult to offer true judgment on whether the curriculum is leading or lagging the demands of the labor market.

As unobtrusive archival sources of information, student records provide rich and complex data that signal changes in a discipline. But they cannot be considered in isolation from other sources. They offer a breadth of variables covering generalized subject matter, student participation, and performance. But they do not offer depth or certainty in matters of expectations for student learning in the field. Official guidelines of professional associations (for example, Tucker et al, 1991) provide indicators of ideal curricula, and tracking changes in the content of those guidelines would be a helpful addendum to understanding what the field considers to be its leading pedagogical edge.

Part 3: The Story Told by Examinations

National examinations dealing with specific subject matter, whether official or de facto, are statements of expectations for summative student learning in a field at a particular transition point in education. In the U.S. system of education, where nothing is official, these examinations are used principally in the process of selecting students into college or graduate programs. As such, the assessment of examination content for an undergraduate field of study is inseparable from consideration of changes in the volume, level, and shape of graduate programs.

3.1 Expansion and Consolidation at the Graduate Level

Let us thus consider, first, what happened to graduate programs in computer science between 1984 (the first year of mass marketing and accessibility of personal computers) and 1993. Table 8 displays considerable growth and contrary tendencies of differentiation and consolidation in the field. There are three major trends evident in the data.

First, a stronger paradigm and control at the doctoral level. The number of Ph.D. programs increased by 40 percent, and the proportion of those programs requiring a first degree in computer science also grew—from 11 percent to 17 percent. At the same time, the number of Master's programs rose more slowly (by 18 percent), with the proportion of those requiring a first degree in computer science falling from 17 percent to 13 percent. Programs, though, are not students, and program requirements do not describe actual student behavior. While the proportion of graduate programs in computer science requiring an undergraduate degree in the field is less than 20 percent, the proportion of prospective computer science graduate students with an undergraduate degree in the field doubled to nearly 60 percent between 1978 and 1987 (Grandy and Robertson, 1992, p. 56).

Second, consolidation. The number of Ph.D. programs offering degrees in more than one field of computer science (such as systems analysis) declined so that the ratio of fields to programs contracted from 2.7 to 1.7. In an age and environment (universities) in which specialization is an indicator of intra-organizational status (Bresser, 1984) and a by-product of faculty belief systems (Simsek and Louis, 1994) and in which competing graduate programs market themselves by subdivision (Geiger, 1993; Smith, 1985), this trend seems counter-intuitive. But on a continuum of computer-related field degrees at the bachelor's level, ranging from hardware oriented (computer engineering) through software oriented (computer science and information science/systems) to management oriented (management of information systems) subfields (Turner, 1993), the doctoral programs moved toward the center, in computing, an indication of focus on research in the core underlying field. At this stage in the evolution of computer science, the outlines of a clear pyramid emerge. In fact, the surveys of degree recipients conducted by the Computing Research Association (CRA), a body established in 1972 to facilitate communication among academic and industry personnel engaged in research in the computer science field, use only two categories, computer science and computer engineering, and 110 of the 149 departments surveyed in 1992–1993 were entitled simply Computer Sciences or Computing Sciences (CRA, 1994; CRA, 1995).

As a by-product, the third trend becomes visible: establishment of a clear distinction between the scholarly and pedagogical canons. The major degree field changes at both Ph.D. and Master's levels include declines in programs in computer programming, systems analysis, and data processing, and the opening of programs in microcomputer applications and "other" fields, usually interdisciplinary, such as human-computer interactions. These changes are yet another sign of a maturing discipline that has developed stronger first level degrees in core subfields, has moved its early research into the undergraduate classroom (the pedagogical canon) and can rely on its Ph.D. programs to produce a scholarly canon. The

Table 8. —Growth and change in graduate programs: computer science, 1984–1993

	1984–85	1992–93
Total Doctoral Programs	85	119
Requiring GRE Subject Test	51 (60%)	40 (34%)
Requiring 1st Degree in Computer Science	9 (11%)	20 (17%)
New Programs within 2 yrs.	5 (6%)	7 (6%)
Fields:		
Computer Sciences	84	78
Information Sciences	44	48
Computer Programming	34	27
Systems Analysis	34	21
Data Processing	25	6
Microcomputer Applications	—	12
Other	—	15
Total Master's Programs	128	151
Requiring GRE Subject Test	26 (20%)	17 (11%)
Requiring 1st Degree in Computer Science	22 (17%)	20 (13%)
New Programs	28 (22%)	15 (10%)
Fields:		
Computer Sciences	108	105
Information Sciences	48	21
Computer Programming	25	9
Systems Analysis	26	12
Data processing	25	6
Microcomputer Applications	—	9
Other	—	21

SOURCES: Graduate Record Examinations Board, 1983 and 1993.

Master's programs also served to explore and establish advanced canons at the software oriented and management oriented portions of the computer/information science subject continuum (Eerkes, 1991).

In this light, it is at first surprising that, as table 8 shows, the proportion of graduate programs requiring the Graduate Record Examination subject test in computer science declined for both Ph.D. programs (from 60 percent to 34 percent) and Master's programs (from 20 percent to 11 percent). It could very well be that these trends are related to the proportion of graduate students in computer science from other countries[6], but the examination itself is constructed with reference to domestic practices. The content of the subject matter examinations administered by the Graduate Record Examinations Board (GREB) is set by advisory committees drawn from the disciplines, and is periodically reviewed to ensure that the examinations are assessing current generalizable content of baccalaureate programs (Adelman, 1989). We assume that faculty experiencing consolidation, differentiation, and a clarification of the pedagogical canon ensure that such examinations reflect what they are teaching, and will use the examinations with confidence. But that is not always the case.

3.2 The Lagging Content of National Examinations

There was a time when the GREB conducted "content representativeness" studies of the tests. Computer science was the subject of such a study only once, in 1982 (Oltman, 1982). If we look carefully at table 9, we note that, at the time, with the exception of one major area (computer organization and architecture) there was considerable mismatch between the content of the examination (the "official committee guidance" column) and the faculty's description of the current curriculum in the field. The faculty's "ideal curriculum" in terms of distribution across the five major areas was much closer to the test specifications.

Furthermore, if we compare the detailed descriptions of topics within the major content categories of the GRE subject test in computer science in 1982 (Educational Testing Service, 1982) and 1986 (Graduate Record Examinations Board, 1986) there was absolutely no change during a period in which the personal computer industry experienced its greatest expansion, and the proportion of undergraduate degrees in computer science rose to what have subsequently proved to be all time highs (Snyder, 1995). At best, then, the content of the examination—the generalizable portion of the undergraduate curriculum in computer science—could be described as "concurrent." Even when versions of the Graduate Record Examination subject tests that were designed for the assessment of undergraduate program effectiveness (called "major field tests") were developed in the early 1990s, the content specifications in computer science were identical to those of the GRE examination of 1986.

At second glance, then, the decline in the proportion of graduate programs requiring the GRE subject test in computer science is not surprising: the faculty judged the relationship of the test to the field as somewhat askew. Indirect support for this position can be found in the results of a 1984 survey of "leading graduate departments" (Bowen and Schuster, 1986).

Table 9.—Faculty perceptions of content of the "national" summative examination in computer science, 1983

Content Area	Faculty's Current Curriculum	Faculty's Ideal Curriculum	Classification of Test Items	Official Committee Guidance
1) Software Systems and Methodology	40.7%*	33.1%	33.0%	**35.0%**
Data Organization, Program Control, Programming Languages, Systems				
2) Computer Organization and Architecture	20.1	20.7	20.1	**20.0**
Logic Design, Processors and Control Units, Memories, Network Structures/Protocols, Pipelining, Synchronization				
3) Theory	12.1*	17.7*	20.5	**20.0**
Automata and Language Theory, Analysis of Algorithms, Complexity, Correctness of Programs				
4) Computational Mathematics	15.5*	16.6*	23.4*	**20.0**
Discrete Structures, Combinatorics, Numerical Mathematics, Linear Algebra				
5) Special Topics	11.5*	11.9*	2.7*	**5.0**
Modeling, Simulation, AI, Computer Graphics, VLSI, Data Communications, Data Bases				

*Differs from Committee Specifications, $p \leq .05$

SOURCES: Oltman, 1982; Grandy, 1989.

Faculty were asked to rate changes in the quality of "advanced graduate students" over a 16 year period (1968–1984). Of 32 fields, computer science ranked fourth highest in student quality change. But during the period 1977 (when the GRE Advanced Test in Computer Science was introduced) and 1984, scores on that test declined—if modestly[7]—providing some evidence of dissonance. On the other hand, it could be argued that, because computer science draws undergraduates from many fields, a single achievement test score is not a valid indicator of potential, and does not reflect the diverse prior learning of students who seek graduate degrees. Put another way, as Oltman and Hartnett found in a 1981 survey, graduate departments (in all disciplines) that do not use GRE scores simply feel that their existing selection criteria and processes are adequate (Oltman and Hartnett, 1984).

In 1994, the Graduate Record Examinations Board conducted a survey of 141 graduate computer science departments to determine various factors in the use of the examination in graduate school admissions. Of the departments that did not require, recommend, or use the test for selection, content related reasons were cited by approximately 40 percent of the respondents (GREB, Personal Communication, July, 1996). What we conclude from this is that the de facto national examination is not a productive indicator of information about either content or change in the curriculum as experienced by baccalaureate students.

3.3 The Promise of Other Examinations

We have reached a stage of technology at which examination problems can be developed and shared electronically across many institutions and, indeed, national borders. To the extent to which faculty can agree on the form and objectives of these problems, it is possible to develop truly international indicators of competence in the field.

In illustrating the ways in which a "mastery examination program" in computer science might work, Grandy (1989) emphasized common conditions and schema. She posits a situation in which examinees are given five hours on a computer to write a program addressing one of nine comparable problems (randomly selected from a pool) that require them to

> First . . . decide on a data structure. Second . . .write similar routines . . . To obtain a minimal passing grade, they must write a program that performs both simple I/O and sequential search. They obtain higher grades by successfully modifying records, modifying the data structure by inserting and deleting records, and by sorting the structure. (Grandy, 1989, p. 35)

The logistics of the examination require the student to record all interactions and to execute the program. The "expert system" that judges the entire production yields a descriptive report, not a "score." But this descriptive report could be based on *types* of errors or *types* of modifications to a program, even if the task involved a "best-average-worst" cases scenario (a typical setting for computer science problems).

Not all on-line problems involve programming to be judged by expert systems. The emerging field of human-computer interactions offers some cases in point. Based on a sample final examination from the University of Toronto in Canada (ACM, 1992), students in many universities in many countries could be given a description of a user interface with either a specific piece of hardware (e.g. printer) or a generic software program (e.g. spreadsheet), asked to critique the interface in terms of "constraints, mappings, conceptual models, visibility, and feedback" (ACM, 1992, p. 152), and to redesign the interface to mitigate those problems, with an accompanying discussion of trade-offs. It is difficult to identify competence in human-computer interactions on the basis of student records, though sooner—rather than later—we will observe enough empirical evidence of study in the field to open new course categories in the taxonomy (section 2.1). But indicators of learning—and in emerging subfields of the discipline—will be easier to construct through content analysis of examinations.

3.4 Beyond Disciplinary Content

From a labor market perspective, one problem with using subject matter examinations as indicators is that it is difficult to tease out of them the kinds of generic learned abilities that provide individuals with flexibility in the workplace. Even if we focus on reasoning abilities, studies have shown that computer science faculty rate particular kinds of reasoning—for example, problem definition, constructing arguments, and avoiding specific fallacies—as more salient expectations for their students than other types of reasoning skills (Powers and Enright, 1986; Fung, O'Shea, and Goldson, 1993). These abilities, critical to creativity and productivity in computer-related occupations, are rarely assessed in large scale subject matter examinations. When national committees set specifications for examinations, they would rather students demonstrate knowledge of sequential circuits, parsers or UNIX. Even these specifications may lag the current demands of the field as practiced in the labor market.

Part 4: The Story Told by Cohort Labor Market Experience

Labor market information is directed as much to the future as to the past. The basic question for higher education is not whether a course of study prepared students for today's labor market, rather whether it will prepare them for the work force of the year 2005.

Conventional wisdom says that computer-related occupations experienced exponential growth in the U.S. labor market over the past two decades. Conventional wisdom is not wrong. But the data show that this growth was uneven and complex, and did not affect all industries. In fact, as Kutscher notes, the measurements themselves are subject to question due to the methodology used "to capture and exclude the price effects related to rapid changes in technology" during the period, 1975–1990 (Kutscher, 1992). National Science Foundation surveys of the employment of computer personnel in industry (Gannon, 1992a; Gannon, 1992b; Morrison, 1995) also reveal that large sectors of the U.S. economy, for example, agriculture, mining, construction, food processing, and textiles, employed comparatively few computer workers by the end of that period.

Nonetheless, by 1986, work force growth rates for computer specialists were the highest of any field of science and technology in the United States, and employers in both manufacturing and non-manufacturing sectors of the economy were paying considerable premiums for advanced computer training and skills at all degree levels (Lane, 1988). The annual employment growth rate between 1976 and 1986 averaged 17 percent and the total number of employed computer specialists (analysts, engineers, programmers) rose 372 percent to 562,600 workers (Lane, 1988, table 1). The demand was such that only five percent of recent bachelor's degree recipients in computer science in 1984–1985 were full-time graduate students, the lowest ratio of all science and engineering fields (Lane, 1988, table 6), a ratio that held through the late 1980s (see section 4.1, below). Reinforcing this trend, only seven percent of computer specialists were employed by educational institutions, the lowest ratio of any science and engineering field (Lane, 1988, table 9).

College students often make program and career decisions according to concurrent labor market trends (Clarke and Teague, 1996), and, whether or not they major in computer science, will read trade press headlines for hints of what to study: "Damn, They're Hot: C++ Programmers" (Haber, 1996); "Steaming Brew: Demand for . . .JAVA Programmers" (Wilde, 1996); "Object Programmers: Hard to Find . . ." (Gaudin, 1996). College-bound high school seniors are even less informed, and the kind of student drawn to the glitz of the labor market may not be the best bet to complete a degree. The history of the High School and Beyond cohort verifies the subjunctive. Of those who, as seniors in high school, planned to major in specific fields and subsequently attended college, the proportion of computer science "wannabees" who actually earned a bachelor's degree was comparatively low:

Proportion of College-Bound High School Seniors Planning to Major in Selected Fields Who Earned Bachelor's Degrees in Any Field by Age 30:
(standard errors in parentheses)

Planned Major	Proportion Earning Bachelor's Degree
ALL	**44.5%**[8]
Physical Science/Mathematics	65.3 (5.35)
Biological Sciences	62.4 (5.87)
Social Sci Other than Psychology	60.6 (2.85)
Humanities	58.8 (5.16)
Communications/Journalism	57.4 (4.40)
Engineering/Architecture	55.5 (2.62)
Psychology	54.0 (6.02)
Education	50.7 (4.38)
Business	43.4 (2.08)
Nursing/Health Sciences	42.0 (4.13)
Computer Science	40.7 (3.07)

Furthermore, among those who planned majors in computer science *and* earned bachelor's degrees, only 32 percent of those degrees were in computer science or computer engineering (another 31 percent were in business fields). The career plans of late adolescents, often based on perceptions of market characteristics such as job share and wages (Fiorito and Dauffenbach, 1982), may not be very stable in emerging occupational areas, and, in the early 1980s, computer-related occupations were imperfectly understood. On the other hand, of those who actually earned bachelor's degrees in computer science, 53 percent had planned to major in the field as high school seniors; another 23 percent had planned to major in engineering; and (not surprisingly) these students had a far stronger secondary school mathematics background than those who evidently did not understand the nature of computer-related occupations or the path to those occupations.[9]

Let us keep our focus on the bachelor's degree recipients. What tracks did these students make in the work force? Into what industries and occupations did they actually walk? Were they prepared for their principal work activities? These questions draw us back to the interaction between changes in the knowledge content of work and the shape of delivered knowledge in higher education.

4.1 Early Career Placement and Graduate Education

There are a number of ways to look at the labor market in relation to a specific field of education and training. The first of these uses the trained individual as the unit of analysis and asks where these people go in the labor force after their requisite years of schooling have concluded. Drawing on data from a National Science Foundation survey (Tsapogas, 1992), table 10 illustrates this approach for one group of recent bachelor's degree recipients in computer science (in 1988) and asks where they were two years later. For our purposes, this group is very convenient since the date of graduation, 1988, is very close to the median date of graduation of the 1982 cohort in the student records analysis, 1987. To be sure, the portrait of labor market status two years after graduation is incomplete, but it is a better indicator than one year and provides potentially convincing evidence of coherence between education and training and job duties. Under this vision, the individual's education and training constitute a significant portion (though hardly all) of what the human resource development literature calls "bio-data," and bio-data is highly regarded as a predictor of job performance (Owens, 1976; Howard, 1986). Our early career histories do not include direct assessments of performance, but they do show us where the knowledge and skills content of the "bio" goes in terms of occupation and duties.

Table 10 shows us, first, that 83 percent of those who were neither full-time graduate students or unemployed/out-of-the-work-force (the "net work force" category) were employed in science and engineering occupations by 1990, and 77 percent in computer science occupations. More notable is the fact that one out of five were working in development and more than two out of five were involved in statistical reports and on-line computing as primary work activities.[10] As one would expect of entry-level placements, very few were involved in research or the management of research and development. If we think back to

Table 10.—Tracking computer science graduates of 1988 into the labor market: bachelor's level

	Numbers of Graduates and Workers	
A) Total # of Graduates, 1988	34,500	
Full-time graduate students, 1990	-1,700	
Number in other activities, 1990	-200	
B) Net # in Labor Market, 1990	=32,600*	
Unemployed/Out of Labor Force	-1,100	
C) Net # in Work Force, 1990	=31,500	
Employed in Science & Engineering Occupation	(26,300)	
Employed in Computer Science Occupation	(24,200)	
D) Primary Activity of Net Work Force, 1990		**Percent of net work force (C)**
Applied Research	400	1.3%
Development	6,400	20.3
Management of R&D	400	1.3
Management of Non-R&D	2,300	7.3
Teaching	700	2.2
Production	2,800	8.9
Statistics, Computing	13,800	43.8
Sales	600	0.2
Professional services	100	0.03
Other	3,700	11.7
No Report	300	1.0

* Of whom 3,800 were part-time graduate students.

SOURCE: Tsapogas, 1992.

the analysis of student records, the content skills most necessary at these early occupational stages are those associated with systems analysis and software, advanced programming (e.g. compiler languages), graphics, and English composition. Referring to table 6, it appears that the students in the 1982 cohort were well prepared for this type of work.

Table 10 also shows us that 16 percent of the 1988 bachelor's degree recipients continued on to graduate school, though the bulk of them did so on a part-time basis while employed. These particular data do not tell us the precise nature or field of the graduate program. But from the longitudinal age cohort studies of the National Center for Education Statistics we know that the percentage will rise in time: for the 1982 cohort, 29 percent of the bachelor's degree recipients attended graduate school (in formal programs, not post-baccalaureate continuing education) by age 30 (1993). For computer science majors, the figure was 22 percent, and their degree attainment and fields are shown below.

Table 11.—Post-graduate work of computer science bachelor's degree recipients, 1982 Cohort, by Age 30 (1993)

Type of Post-Graduate Work		
None:	68.1%	
Continuing Education Course Work Only:	10.0	
Business Field:		4.4%
Computer Science:		4.7
Other:		0.9
Incomplete Graduate Degree:	5.7	
Business Field:		3.0%
Computer Science:		0.7
Mathematics:		2.0
Master's Degree:	16.0	
Business Field:		6.7%
Computer Science:		7.3
Communication Technology		2.0
First Professional Degree:		
Law	0.2	

The overwhelming content of the post-baccalaureate work of computer science majors in this cohort, then, was either in business or computer science. And while the analogy is hardly perfect, the proportion who started but did not complete degrees is in line with Bowen and Rudenstine's report of attrition rates in doctoral programs in mathematics and physics (the closest fields to computer science in their study) in ten institutions (Bowen and Rudenstein, 1992). The match with industry employment patterns was well made. Using a sample of older 1992 GRE test takers, Grandy (1997) found that the "holding power" of computer science was even greater than that implied in this table, with nearly 75 percent of returning computer science majors planning graduate work in the same field. For our High School and Beyond cohort, then, it may be that some of that 78 percent who had not gone on to graduate school by age 30 will return in time, and in computer science.

4.2 Looking Back: What They Brought to the Labor Market

We can pause at this stage of the path, and ask a question that looks backward to the contents of undergraduate education: what, specifically, have people working in a given occupational cluster studied? If we take those in the HS&B/So cohort who were working in computer-related occupations (other than computer engineering) in 1991 (age 28) and ask

Table 12.—Proportion of 28-year-old workers in computer and other technical occupations in 1991 who had completed undergraduate courses in sixteen aggregate categories

Aggregate Course Categories	Computer Occupations	Other Technical Occups.
Composition and Writing	83.0%	75.6%
"College-Level Math"*	67.9	56.5
Computer Sci. (other than programming)	67.4	34.8
General Psychology	60.8	42.3
Computer Programming	59.3	24.2
Business & Management: General	58.4	34.1
Calculus & Advanced Math	53.5	31.0
Oral Communication/Speech	51.9	39.0
Introduction to Economics	51.5	44.1
Literature/Letters	49.5	45.0
Data & Computer Applications	47.5	24.5
Accounting	46.6	20.4
U.S. History/American Civilization	42.5	35.9
Philosophy/Religious Studies	36.7	35.9
Mathematical Statistics	36.6	26.4
Physics	34.2	39.2

*Includes college algebra, trig, precalculus and finite and discrete mathematics.

what proportion of the group successfully completed courses in various fields, we can begin to draw a portrait of the knowledge and skills they brought to the labor market. Table 12 uses 103 aggregate course categories and selects those categories in which more than one-third of the population who eventually worked in computer-related occupations earned credits, and compares this profile to the enrollment proportions in the same course categories by those working in other, non-health related technical occupations (excluding scientists and statisticians) in 1991. The population, in both cases, includes people who earned associate's degrees, bachelor's degrees, and no degree but 45 or more credits.

This is obviously a different way of looking at the relationship between the content of higher education and the labor market. The knowledge portrait of computer analysts, programmers, software support specialists, Management Information Systems managers, and network administrators presented above should be very encouraging to any industry. The principal emphasis of content is where it is supposed to be: computer-related fields and mathematics. But the portrait is very balanced with business, humanities, and communication skills. The category of other technical occupations (such as engineering assistants, environmental technicians, navigators, laboratory research associates) presents a more diffuse curricular profile, even though those with an engineering technology background constitute a significant proportion of the whole.

4.3 Industry Analysis

A third way to examine the track from education to the labor market is to use a combination of industry and occupation as a guide to the current status of the work force. Tables 13 and 14 set forth this grid for both manufacturing industries in 1989 and 1992 (table 13) and non-manufacturing industries in 1990 (table 14). To be sure, this account does not include "computer personnel" who worked for government, the military, or educational institutions, for example. But 85 percent of computer personnel are employed in business and industry, principally in manufacturing and service sectors (Tsapogas, 1992). It may well be the case that some of these personnel are working on projects funded under contract to government or the military, but their employer is a private sector entity.

Tables 13 and 14 are not exactly parallel. In terms of general occupation, the two presentations hold computer analysts and computer programmers in common. But they also illustrate the principal problem with labor force data in computer-related fields: the counting of computer engineers. In some analyses (e.g. most of the 1992 OECD country reports), they are lumped with electrical engineers. In others (for example, Gannon, 1992b), the industry cells are so small that all engineers are lumped together. It is difficult to assess the distribution of computer engineers in the manufacturing economy when the occupational category, "other computer specialists," is ambiguous. In fact, the presentation of "other computer specialists" in table 13 has to shift between an aggregate category and computer engineers in order to report employment level changes within industries for the two years at issue.

Table 13.—Employment of computer personnel in selected United States manufacturing industries, 1989 and 1992

Industry	Computer Analysts		Other Computer Scientists		Computer Programmers	
	1989	**1992**	**1989**	**1992**	**1989**	**1992**
All	102,800	106,000	Not Ind.	77,300*	91,600	71,300
Computer & Office Equipment	10,640	10,320	19,000	32,370	26,890	14,300
Aircraft & Parts	12,640	17,750	2,210	N.A.	5,730	4,760
Navigation Equipment	7,110	5,790	3,430	7,340	3,000	2,210
Electronic Components	4,040	5,110	6,000	11,460	4,480	3,140
Space Vehicles, Missiles	3,980	3,900	1,690	1,610	2,840	2,930
Communications Equipment	2,330	3,360	3,980	5,920	3,360	2,040
Drugs	3,210	5,510	320*	530*	1,490	1,680
Measurement Devices & Controls	2,630	2,350	3,720	6,020	2,220	2,540
Motor Vehicles	2,830	2,350	350	700	1,540	1,570
Plastics & Synthetics	1,620	1,550	320*	250*	630	420
Newspapers	1,350	1,640	340	N.A.	1,170	1,250
Construction Machinery	1,600	2,150	550*	550*	720	990
Medical Instruments	1,180	1,530	740	1,840	900	1,250
Periodicals	1,070	1,270	120	N.A.	1,100	990
Books	1,250	1,640	100	N.A.	780	570
Organic Chemicals	1,480	1,110	150*	210*	770	640
Petroleum Refining	1,060	940	450*	590*	390	910
Basic Steel Products	1,110	1,360	610*	630*	540	420

*Computer Engineers only.

SOURCES: Gannon, 1992b.; Morrison, 1995.

Table 14.—Employment of computer personnel in selected United States non-manufacturing industries, 1990

Industry	Computer Analysts	Computer Engineers	Computer Programmers
All	203,700	62,600	238,100
Computer & Data Services	80,890	40,220	117,360
Commercial Banks	19,050	1,280	11,910
Life Insurance	13,800	730	12,060
Research & Testing Services	7,380	6,660	8,810
Engineering & Archit Services	5,380	7,920	7,530
Fire/Casualty Insurance	8,020	810	9,250
Medical Services/Insurance	6,850	170	4,530
Accounting & Bookkeeping	6,120	360	3,980
Personnel Supply Services	2,510	2,120	5,480
Stock Brokers	4,350	190	4,900
Insurance Agents & Brokers	3,850	300	4,500
Crude Petroleum/Natural Gas	3,180	230	2,120
Financial Savings Institutions	2,740	150	2,230
Miscellaneous Business Services	1,280	310	3,720
Advertising	1,010	40	440
Mortgage Bankers	1,050	40	990
Hotels and Motels	860	260	540
Oil & Gas Field Services	410	160	560

SOURCE: Gannon, 1992a.

Even then, data may be not be available, and for reasons that would not be apparent to the casual reader. For example, in 1989, there were 2,210 "other computer specialists" in the aircraft manufacturing industry. In 1992, these data disappear. One discovers the reason only by discussing the phenomenon with personnel at the Bureau of Labor Statistics (BLS), which conducts these surveys for the National Science Foundation. The aircraft manufacturing industry is an oligopoly. Firm A employs X number of computer specialists. Using BLS data in combination with other information, firm A could determine how many computer specialists were employed at firms B and C. For that reason, it was agreed that the information concerning computer specialists in the aircraft industry was proprietary and no data would be published.

While employment levels of computer personnel in some manufacturing industries remained relatively stable in the short term covered by the NSF reports for 1989 and 1992 (see, for example, the lines for space vehicles/missiles, plastics & synthetics, and basic steel products), the data in most industries exhibit a degree of volatility that stretches the boundaries of credulity. It could be the case that the same people are shifting job categories and titles within the same industry, for example, from programmer to analyst. But BLS personnel point out that there is a great deal of "residual addition" in the construction of these data reports, that is, whatever cannot be accounted for in major categories of analysis is imputed.

Nonetheless, we can conclude that the non-manufacturing sector employs the majority of computer personnel, and that, within the non-manufacturing sector, financial services (banks, insurance, brokerage, and accounting firms) are second only to computer and data service firms themselves as "destinations" for computer personnel. It is no wonder that, in the 1982 age cohort, computer science students at both bachelor's and (particularly) associate's levels devoted measurable undergraduate time to the study of accounting, business administration, and business computer applications.

We can also conclude that while there is an overall balance between analysts and programmers, the mix differs by industry. Programmers dominate employment in the computer industry itself, both manufacturing and non-manufacturing portions. But in other industries (e.g. aircraft, navigation equipment, and commercial banking), analysts dominate. Students preparing for changing labor markets can use this information to add appropriate course work to their programs. We can use this information as a gloss on student records to see whether computer science students at either associate's or bachelor's levels have appropriate knowledge in their valises. For example, if, as table 13 shows, the publishing industries (newspapers, periodicals, books) are major employers of students trained in computer fields, do the students have any academic background in either journalism or graphic communications? If they do, they will adapt more quickly to their industry environments and evidence a higher degree of productivity. But in both cohorts of our national transcript samples, computer science majors at associate's and bachelor's degree levels completed virtually no courses in either journalism or graphic communications. On the other hand, 10 percent of the bachelor's degree recipients working in computer-related

occupations majored in the humanities or arts, and hence can cover at least some of the knowledge and skills needs of those industries.

4.4 Occupational Growth Projections

There are many worthy reasons for continuing one's education beyond secondary school. Developing knowledge and skills to establish a career is only one of them, but it is the objective of greatest concern to national investments in higher education—let alone to many students.

One key to student choice in terms of level and content of degrees is the occupational outlook in the student's field of interest. Because computer-related occupations are scattered through a wide range of industries, occupational projections are based on a complex analysis involving

- changes in industries, such as production technology,
- occupational structural changes, such as staffing patterns, and
- net replacements, due to projected retirements

among other factors. The best an analyst can do is to offer three levels of projections for an occupation: low, moderate, and high (see, for example, Silvestri and Lukasiewicz, 1992, table 2). The results of these projections are, on the whole, accurate. With the exception of the category of computer equipment repairers, the 1982–83 Bureau of Labor Statistics projections for 1990 employment in computer-related occupations were, at worst, slightly understated (Rosenthal, 1992). The demand for computer equipment repairers declined as a function of improvements in both materials and manufacturing technology, developments that were not foreseen when the projection was initially made.

Table 15 focuses on the most intense interaction of occupation and industry for computer-related occupations, namely the computer service industry itself. It is a fairly large industry (831,000 employees in 1992) with a projected growth rate to 2005 across all occupations of 95.7 percent. But within the industry, there are considerable variances in projected occupational growth rates, from a low of 42 percent for secretaries, clerks, and other administrative support personnel to 216 percent for systems analysts. The number of programmers is predicted to increase at a lower rate than other occupations in the industry, but this trend is related to an overall projected slowing of the rate of growth in what is already a large occupation. Even in recent years, actual declines in the employment of programmers in industries that adapt quickly to large scale software and communication systems have been documented (Graff, 1995).

Table 15.—Projection of growth in selected occupations: United States computer service industries, 1992–2005

Occupation	1992 Employment Number (000)	Percent	1992–2005 % Change
All occupations	**831**	**100.0%**	**95.7%**
All Administrative Support Occupations	231	27.8	42.2
Professional Occupations	**191**	**23.0**	**188.8**
*Systems Analysts	74	8.9	216.0
*Computer Engineers & Scientists	56	6.7	216.0
*Electrical Engineers	16	1.9	102.8
*Operations Research Analysts	7	0.9	97.5
Technicians and Programmers	**163**	**19.6**	**80.7**
Computer Programmers	140	16.9	77.8
Engineering Technicians	20	2.5	102.0
Managerial and Executive	**142**	**17.1**	**87.1**
*General Managerial/Executive	39	4.7	52.7
*Engineering, Mathematical and Natural Science Managers	24	2.9	110.1
Marketing and Sales	**59**	**7.2**	**70.4**
Repair and Precision Production	**32**	**3.8**	**93.9**

* Occupations for which a bachelor's degree is usually required.

SOURCE: Bureau of Labor Statistics, 1994

4.5 The Anomaly of Credentials

There is no question that the highest intra-industry growth rates will occur in occupations for which a bachelor's degree is required. But, as table 16 demonstrates, computer-related occupations are anomalies in terms of the formal educational attainment of workers. Among

Table 16.—Educational attainment of full-time workers in selected occupations, 1992

	Highest Degree Earned					
OCCUPATION	**<BACH**	**BACH**	**MAST**	**1st PROF**	**PH.D.**	**000s**
Computer System Analysts	16%	58%	24%	<1%	2%	576
Operations Research Analysts	41	40	16	0	3	183
Computer Science Professors	8	17	42	0	33	12
Chemists	17	43	13	3	24	120
Chemistry Professors	0	6	18	<1	76	17
Biological Scientists	9	42	26	5	18	84
Biological Science Professors	0	4	9	4	83	23
Psychologists	13	10	42	9	26	158
Psychology Professors	0	6	13	6	75	16
Accountants	31	57	12	<1	<1	1,264
Lawyers and Judges	0	4	9	73	14	735
Social Workers	28	47	24	<1	<1	536
Nurses	50	42	7	1	<1	1,332
Editors, Reporters	28	57	13	<1	2	269
Architects	13	56	25	6	0	126

SOURCE: U.S. Bureau of the Census, 1992, Tables 30076 and 30077.

college professors, for an obvious example, the proportion in computer science who hold the Ph.D. is less than half of the rate in chemistry, biology, or psychology. In fact, 25 percent of computer science faculty hold no degree higher than the bachelor's.

The computer field has honored competence, content, and creativity more than credentials, and in this respect it has presented a significant challenge to the putative structure of rewards in the intersection of education and the labor market. A cultural hypothesis lurks here, one too large for this excursion. A hyperbole will serve the point: Bill Gates was a college drop-out; everyone knows that fact; and there are 100,000 would-be clones waiting in line.

We know from previous studies of the 1972 cohort (Adelman, 1994) that where students took higher level mathematics study into managerial roles in selected industries (financial services, communications, durable goods), their productivity (measured by wages) was enhanced. Degree completion was a secondary issue. It can be hypothesized that the supply of quantitative skills brought by these students to their managerial roles helped those industries adapt more efficiently to the changing technological environment of the early 1980s. Will history repeat itself in the story of the 1982 cohort? At age 28–29, it is too early to tell. But when and if we are able to ask the question, various types of computer science courses (programming, applications, architecture) will be added to mathematics. It is a natural progression.

4.6 Confirmations of Cohort History

What happened to the members of the high school class of 1982 who subsequently earned bachelor's or associate's degrees in computer science? The taxonomy of occupations and industries developed for the penultimate year for analyzing employment in the High School and Beyond data base, 1991, was arrived at in the same empirically inductive manner as the course taxonomy. That is, we had records of 1992 telephone interviews with respondents that indicated their answers to the following questions:

- What was your most recent occupation?
- For whom did you work? and what kind of organization is that?
- What were your principal duties on the job?

Because we had the "literals" (key words the respondents used in their telephone interviews) for job duties, along with the respondent's educational history, we could discriminate far more accurately and with sensitivity to different levels of computer-related work. We used, as initial guidelines, the Census categories previously employed for the high school class of 1972. But there are simply too many categories in the Census taxonomy to yield statistically significant comparisons for small populations. After sorting responses into a more finite set of bins, and adjusting the bins, 40 job categories resulted, including:

- Data Entry Clerks (distinguished from three other categories of clerks)
- Computer and Computer Equipment Operators
- Computer Programmers
- Technical Workers: Computer-Related, Other than Programmers (e.g. systems analysts, software support specialists, MIS managers, etc.)

Computer and software engineers were included under the umbrella of "engineers," with the requirement of a bachelor's degree. Someone who claimed an occupational title of "engineer," described their duties as "hardware technical support," and did not possess a bachelor's degree, for example, was classified as a "technical worker: computer-related," not an engineer.

Two-thirds of the 1982 high school graduates who subsequently received bachelor's or associate's degrees in computer science and related fields were in computer-related occupations in 1991.[11] None of the bachelor's degree recipients was working as either a data entry clerk or computer operator. The proportions in computer-related and non-computer-related occupations, by degree level was as follows:

	Associate's Degree	Bachelor's Degree	% of All
All Graduates	21.4%	78.6%	—
In Computer-Related Occupations	17.2*	82.8*	67.1%
In Non-Computer-Related Occupations	29.8*	70.2*	32.9

*$p \leq .05$

Associate's degree recipients, then, were less likely to be employed in computer-related occupations, but this does not mean that their learning was lost. Credit examiners, sound engineers, or computer operators with associate's degrees in computer-related fields carry knowledge in their valises that can only expand the boundaries of their work, and give them an advantage in adapting to changing technical environments.

4.6 "Productivity Orientation" and More Traditional Labor Market Outcomes

However, one of the more remarkable differences between people who wound up in computer-related occupations and those in other occupations, regardless of degree level, is in their "productivity orientation." This concept is derived from a series of questions asked in the surveys concerning what individuals value on the job. On one side of the equation are issues such as pay, job security, and opportunities for personal advancement. These are "me" orientations. On the other side are questions about the challenge and importance of work and the opportunity to acquire more education and training. These are oriented toward the job and improvement or work, and are helpful in analyses of potential productivity effects of education, particularly in comparing the workplace histories of women and men (Adelman, 1994). The saga of the high school class of 1972 was extended by the National Center for Education Statistics far enough in time (to age 32-33 in 1986) and included enough questions about actual job activities so that "productivity effects" could be hypothesized, at least in their economic dimensions. But the surveys of the high school class of 1982 (for which computer-related occupational destinations are far more significant) stopped at age 28-29, and did not include questions about job activities. The best we can do, then, is to describe "orientation," not effects. The problem, of course, is that "productivity orientation" reflects *values* that play into a range of qualitative effects of education, both economic (Haveman and Wolfe, 1984) and social, and the interplay of education, values, and adult choice-behavior are difficult to evaluate (Behrman, 1997).

The first step in the creation of the variable was to generate a "productivity orientation" ratio: the average of "job" responses divided by "me" responses. The higher the ratio, the more the respondents valued those aspects of their jobs that potentially improved the quality of work. The ratio was then spread out on a scale and trichotomized: above average orientation to productivity, a neutral score, and a below average orientation to productivity. The measure was taken at age 22-23 (unfortunately, we did not ask enough of these questions at age 28-29 to generate a statistically defensible ratio). As one might imagine, at age 22-23, respondents will provide more "idealistic" answers than at age 28-29, but the trichotomy softens some of the statistical noise from this phenomenon.

Productivity Orientation and 1991 Occupation of Computer Science Students
(standard errors in parentheses)

	Low	Neutral	High
Degree Holders in Computer-Related Occupations	29.3% (6.1)	26.8% (5.2)	38.5% (5.4)
Degree Holders in Other Occupations	22.1 (5.8)	46.1 (9.6)	29.0 (7.5)

For students with associate's or bachelor's degrees in computer science, the proportion with above average productivity orientation was 38.5 percent of those working in computer-related occupations and 29 percent for those working in other occupations. This difference is not only statistically significant, but represents a modest departure from an ideal trichotomy (that is, where any category should have 33.3 percent of the subjects).

Given the way the variable "productivity orientation" was constructed, one must be careful not to over interpret such results. The wage structures and expectations of occupations play a role in respondent attitudes (Berger, 1988; Polachek, 1978). Teachers, for example, do not go into the profession for the money, and 56 percent of the High School and Beyond school teachers in 1991 were in the "high productivity orientation" category. The same is true for artists, musicians, or newspaper reporters: regardless of the degree they attained (and some did not earn a degree at all), the proportion of people in these occupations with high productivity orientation was 57 percent. It is also true that the higher one's attained degree, the higher one's productivity orientation: 58 percent of those with graduate degrees and 47 percent of those with bachelor's degrees were in the highest productivity orientation bin, compared with 29 percent of those with associate degrees and 20 percent of those who continued their education after high school but earned no degree.

Why use "productivity orientation" instead of earnings, unemployment ratios, and job stability in the analysis? It can be argued, after all, that earnings are a proxy for actual productivity, not merely "orientation" to productivity. In deference to that argument, let us look briefly at those other, more popular, indicators of what the literature calls the "labor market outcomes of education" for three occupational clusters: computer-related, other technical, and engineers and scientists. Table 17 presents the key data for the High School and Beyond cohort. The earnings year is 1991, the last year for which we have full labor market information for this cohort. The dollars are 1992 dollars. The universe consists of all respondents who (1) had continued their education beyond high school, (2) indicated an occupation and at least one job with 1991 earnings of more than $5,000, (3) were unemployed for less than 4 months during 1991, and (4) if they were out of the labor force at all during 1991, were out for less than 5 months. The unemployment ratio covers the six-year period, 1986-1991. The universe for the unemployment ratios is larger, consisting of all respondents who continued their education after high school and indicated an occupation and at least one job in 1991. The ratio is that of months unemployed to total months in the labor force over that six-year period. The mean ratios are low; the standard deviations (S.D.) are not.

The table distinguishes by degree level within occupation. The only key data for the interpretation of earnings that are not included are number of years in the labor market and earlier occupations. But if we are taking the measure at age 28–29, these other variables will not be particularly illuminating, as, for example, people with no degree have usually been in the labor force longer than those with bachelor's degrees.

Table 17.—Mean 1991 earnings and unemployment ratios, by highest degree, for High School and Beyond students who continued their education after high school, in three occupational clusters

Occupation Cluster	Mean Earnings in 1991	S.D.	s.e.	Unemp. Ratio 1986–91	S.D.	s.e.
Engineers (inc. Computer), Scientists, Statisticians	$36,408	$ 9,261	$ 977	.029	.094	.009
Computer-Related	31,898	10,752	982	.026	.069	.006
Bachelor's Degree	35,000	11,133	1,289	.028	.060	.007
Associate's Degree	28,032	5,439	1,425	.008	.031	.008
No Degree	26,211	8,563	1,543	.030	.095	.016
Technical, Other Than Computer	27,916	17,079	1,697	.024	.069	.006
Bachelor's Degree	28,745	22,977	3,444	.033	.078	.011
Associate's Degree	27,690	11,237	3,619	.016	.046	.014
No Degree	27,180	10,066	1,466	.016	.062	.009
ALL OCCUPATIONS	**$26,156**	**15,276**	**297**	**.031**	**.093**	**.002**

What do we learn that bears on the principal story lines of this monograph? First, it's important to note that the occupational category of engineers-scientists-statisticians was defined in such a way that everybody in that bin has at least a bachelor's degree, so differentiation of earnings and unemployment ratio by highest degree earned is a moot question. Second, the category of computer-related occupations is relatively homogenous because all the occupations are tied to a specific technology. It includes not only programmers, but also systems analysts, software support specialists, MIS managers, network administrators, data editors, hardware technicians, and computer equipment operators. The category of "technical-knowledge workers other than computer," however, covers a much more varied territory. It includes laboratory technicians and research assistants other than medical, engineering assistants, environmental technicians, air traffic controllers, navigators, pilots, sound engineers, and others. The technologies (let alone the industries with which they are associated) range widely, and those working in these fields were more likely to have held more than one job (57 percent) between age 25 and 29 than those in computer-related occupations (47 percent).

These differences help explain the strength of degree level comparisons in computer-related occupations and the weakness of those same comparisons in the other category of technical workers. There are clear and statistically significant impacts of degree attainment on earnings in computer-related occupations. The same conclusion cannot be reached in the case of "technical, other than computer." In the matter of unemployment ratios, bachelor's degree attainment seems associated with a higher rate of unemployment, though, when one speaks about 2.9 percent, 2.6 percent, and 3.3 percent of total time in the labor force over a period of six years, one describes perhaps two months of unemployment, and these may be attributable to transitional periods between one job and another. It is also possible that higher paid workers are willing to take risks of brief unemployment in order to obtain better positions.

Earnings and unemployment analyses such as these, I think, detract from the story. We are looking at pathways and the behaviors of individuals along those pathways. Productivity orientation can be attributed directly to our subjects. The other labor market history indicators may be attributable to external sources and environmental factors as well as to the choices of individuals. In a labor market in which just-in-time knowledge and just-in-time skills are as highly valued as they are in computer-related occupations (NRC, 1993), ostensible job stability may be an indicator of stagnation and earnings may hide such creative benefits as stock options. Productivity orientation may make the difference.

Part 5. Missing Issues in the Tracking Story

There are many other features of the paths from the study of computer science into the labor market that were not examined in this monograph, principally because the various data sources do not allow a full tracking. Gender issues are a prime example. One could follow women and men through the NCES longitudinal studies, including college transcripts and labor market histories, but the other data sources used in the narrative of this monograph do not discriminate by gender. When they do, as in the case of NSF's surveys of graduate student enrollment that provide a very helpful and wise distinction between full-time first year enrollments and enrollments *beyond* the first year, non-U.S. citizens are not screened out (see, e.g. Huckenpohler, 1992). The failure to disaggregate foreign nationals is a persistent problem in the reporting of gender data at the graduate level, where foreign students comprise over 10 percent of all students in all fields, and among whom men have historically outnumbered women by a *minimum* ratio of 2:1 (Snyder, Hoffman, and Geddes, 1996, table 203, p. 208). Among U.S. citizens, on the other hand, women have been a majority since 1980, and, in 1994, held a 1.3:1 advantage (Snyder, Hoffman, and Geddes, loc. cit.).

As for race-ethnicity, the analytical problem lies in diminishing sample size when individuals are sorted into many cells. For example, if we look at the High School and Beyond students employed in computer-related occupations at age 30 by race and sex, it appears that black women, who constitute 4 percent of college graduates, are well-represented with a 6.5 percent share of the occupational cluster. But the occupational cluster, in turn, accounts for

only 3 percent of the universe. Hence, if we ask for standard errors of measurement, the software program will generate neither an estimate nor a standard error. The underlying N for .065 x .03 is well below the threshold of measurement. On the other hand, when the question concerns the highest level of mathematics studied in high school by men and women who subsequently earned bachelor's degrees in computer science, we have a fairly robust comparison: 55 percent of the women versus 43 percent of the men in this group completed precalculus or calculus in high school! That begins to tell us something about women's self selection into computer science.

There is, of course, a greater limitation to the tracking story. The two NCES longitudinal studies used in this study stop cold when the cohorts were 32–33 and 29–30 respectively. No further follow-up surveys are planned. The third longitudinal study in the series, for the high school class of 1992 (the study started when the students were in the 8th grade in 1988), obviously has yet to run its course: we will not see a college transcript sample until 2002 at the earliest. Both NCES and NSF sponsor other longitudinal studies of bachelor's degree recipients (and the NCES study of the college graduating class of 1993, "Baccalaureate and Beyond," includes transcripts). However valuable, these data sets do not include people who earned less than bachelor's degrees, and also focus on early stages of graduate education and careers. So the cohort stories, and time series analyses, are in midair, and none will get us to mid-life. The occupational data from the Bureau of Labor Statistics and the National Science Foundation will keep marching forward. But the other sources will not.

Part 6: Leading, Lagging, or Concurrent: A Summary

This monograph has tried to examine and illustrate what various sources of information can tell us about the specific knowledge that students acquire in computer science and take into specific occupations in specific industries, what conditions that knowledge in such a fast growing and maturing discipline, and the extent to which we can determine whether the knowledge is at least concurrent with the demands of the labor market.

It concludes that the educational supply is more important and more subject to delineation than the conventional measures of labor market results, and, in the case of computer science, that the judgment of concurrence varies by configuration of work force variables. Nonetheless, there is a logical path from education to work force that empirical evidence verifies.

What did we find along that path? First, that for those who majored in computer science and earned degrees at the bachelor's and associate's levels, the knowledge content of higher education was concurrent, but needed continued monitoring and boosts from professional societies such as the Association for Computing Machinery to stay concurrent. The labor market data, too, suggest that these students were well prepared for the kinds of positions they assumed in early stages of their careers. Second, and in a contrary direction, we found that without more detailed national accounts of curriculum delivery or disaggregation of large course categories such as computer programming, our confidence level in the judgment of concurrence is not robust. Furthermore, in its present form, the de facto national

examination of undergraduate work in computer science cannot be used to raise this confidence level, though the model of on-line mastery examination programs shows promise in keeping assessment current with the leading edges of pedagogy and industry practice.

Leading edges are difficult to divine in computer science: sometimes we can spot them in graduate programs, but they more likely emerge in industry and among the autodidacts, not in the pedagogical canon. It's not as if we could say that colleges were teaching JAVA before the era of mass Internet. Teaching JAVA is a concurrent phenomenon. What leads a field such as computer science is creativity, flexibility, adaptability, and, ultimately, the kind of productivity that changes the boundaries of worklife. Our job in academic environments is to capitalize on the innate creativity of students and to fashion a learning environment that elicits adaptable behaviors. We must also be more self-conscious and deliberate about moving pieces of content in new subfields from graduate schools into undergraduate programs. That is about as close as we can get to leading edges without crossing the boundary of the proprietary interests of industry.

When we speak of "managing the supply" of students in a field such as computer science, we refer not so much to numbers of people studying in the field, rather to the forces that shape delivered knowledge. The degree of coherence in these forces—professional and disciplinary associations, quality assurance bodies, the assumptions of graduate degree programs, and examinations—determines the contours and texture of the path followed by students, and, ultimately, their contributions to the changing knowledge content of work.

Notes

1. The numerical codes used in the *College Course Map* (CCM) course taxonomy are derived from the coding system used in the 1985 version of *The Classification of Instructional Programs*. With each national college transcript sample, these codes are subject to review and revision in light of the empirical evidence. For a description of the process, see Adelman, 1995, pp. 4-20.

2. For students in the high school class of 1982 who continued their education after high school, the highest levels of mathematics attained in high school were:

	All Students	Students with Mini or Micro Experience	Students with Experience in Programs Other than BASIC
Calculus	6.9% (.504)	10.2% (.832)	17.1% (1.36)
Pre-Calculus	6.0 (.475)	9.0 (.825)	11.5 (1.12)
Trigonometry	11.3 (.621)	14.3 (.945)	17.6 (1.41)
Algebra 2	27.3 (.854)	27.8 (1.26)	26.4 (1.63)
< Algebra 2	48.5 (.931)	38.9 (1.34)	27.5 (1.53)

Standard errors are in parentheses.

3. During the period in which the 1972 cohort was attending college (1972-1984), some computer science degrees were awarded by mathematics departments. The ratio of computer science degrees to total degrees for that period may thus be understated.

4. A full appreciation of this phenomenon would require comparisons with degree recipients in other fields, a task that is beyond the scope of this paper. It is worth noting, though, that for *all* bachelor's degree recipients in both age cohorts, the "top 25" courses accounted for approximately 40 percent of undergraduate time (compared to 52-53 percent for computer science students).

5. To be sure, there are other engineering paradigm theories that see computer science and computer engineering as derived, and not primary, fields. See, e.g. Harms,1996.

6. As an indirect indicator, the proportion of Ph.D. recipients in computer science who were foreign nationals rose from 23 percent in the mid-1970s to 48 percent in 1992-93 (CRA, 1994, p. 12).

7. It is difficult to judge changes in test scores on national examinations over a comparatively short period such as 1976–1983. Expressed in Standard Deviation Units, the change for computer science for that period was –.15, a modest decline. At the same time, the number of test-takers rose from 1,357 to 3,813 during that period (Graduate Record Examinations Board, 1984).

8. This means that of all students in the high school class of 1982 who entered any kind of postsecondary institution by age 30, even if they earned no credits, 44.5 percent earned a bachelor's degree. If we changed the universe to those students who (1) attended a 4-year college at any time during that period (the only people who had a chance to earn a bachelor's degree) and (2) who earned more than a semester's worth of credits (that is, eliminating the incidental students), the proportion earning a bachelor's degree was 65 percent (see Smith, T. M. *et al*, *The Condition of Education, 1996*, p. 25).

9. If we compare those who actually earned bachelor's degrees in computer science to those who planned to major in computer science but who entered higher education and did not major in computer science at any degree level (including no degree), the proportions who reached different levels of mathematics in high school were:

Highest Math in High School	Earned Bachelor's Degree in Computer Science	Planned to Major in Computer Science But Did Not
Calculus	28.9%*	10.3%*
PreCalculus	18.8	17.5
Trigonometry	18.8*	15.6*
Algebra 2	23.9*	34.1*
< Algebra 2	9.6*	22.5*

*$p \leq .05$

10. Taken in the context of a very high growth industry, the area of "development" as a primary work activity experienced a 244 percent increase between 1976 and 1986, compared with a 440 percent increase in statistical work-computing (Wilkinson, 1990). Ironically, in light of such rates, "development" was a relatively stable work activity.

11. The only statistically significant comparisons in the following table are on the rows for "Technical: Computer-Related" and "Computer Programmers."

Occupations at Age 28 (1991)
of Bachelor's Degree Recipients in Computer Science
from the High School Class of 1982

Occupation	Men (61%)	Women (39%)	All
Clerical	2.1%	11.2%	5.7%
Protective Service	2.1	—	1.3
Military	4.8	—	3.0
Financial/Business Support	0.6	1.9	1.1
Financial Service Professionals	5.6	3.5	4.8
Buy/Sell	3.6	3.8	3.7
Educators other than School	2.4	—	1.5
Oth Human Service Professions	—	2.9	1.1
Engineers, Software Engineers	12.0	5.6	9.5
Technical: Computer-Related	24.7	33.9	28.3
Computer Programmers	37.7	27.5	33.7
2nd Level Administrators	0.6	8.1	3.5
Other	3.8	1.6	2.9
TOTAL	100.0%	100.0%	100.0%

SOURCE: National Center for Education Statistics, High School & Beyond, Special Analysis File.

Weighted N = 39,738.

REFERENCES

Adelman, C. 1995. *The New College Course Map and Transcript Files: Changes in Course-Taking and Achievement, 1972–1993.* Washington, DC: U.S. Department of Education.

Adelman, C. 1994. *Lessons of a Generation: Education and Work in the Lives of the High School Class of 1972.* San Francisco: Jossey-Bass.

Adelman, C. 1989. "Indicators and Their Discontents," in C. Adelman (ed.), *Sings and Traces: Model Indicators of College Student Learning in the Disciplines.* Washington, DC: U.S. Department of Education, pp. 1–10.

Albers, D. J., D. O. Loftsgaarden, D. C. Rung, and A. E. Watkins. 1992. *Statistical Abstract of Undergraduate Programs in the Mathematical Sciences and Computer Science in the United States: 1990–1991 CBMS Survey.* Washington, DC: Mathematical Association of America.

Albers, D. J., D. O. Loftsgaarden, and R. D. Anderson. 1987. *Undergraduate Programs in the Mathematical and Computer Sciences: the 1985–1986 Survey.* Washington, DC: Mathematical Association of America.

American Assembly of Collegiate Schools of Business (AACSB). 1991. *Standards for Business and Accounting Accreditation.* St. Louis, MO: Author.

Anderssen, E. C. and Myburgh, C. P. H., "The Acquisition of Operating Systems Concepts by Computer Science Students." *Computers and Education*, vol. 19, pp. 309–320.

Association for Computing Machinery. 1993. *Computing Curricula Guidelines for Associate Degree Programs.* New York: Author.

Association for Computing Machinery. 1992. *ACMSIGCHI Curricula for Human-Computer Interaction.* New York: Author.

Barley, S. R. 1992. *The New Crafts: the Rise of the Technical Labor Force and Its Implication for the Organization of Work.* Philadelphia: National Center on the Educational Quality of the Workforce (University of Pennsylvania).

Behrman, J. R. 1997. "Conceptual and Measurement Issues," in J. R. Behrman and N. Stacey (eds.), *The Social Benefits of Education.* Ann Arbor, MI: University of Michigan Press, pp. 17–68.

Beniger, J. R. 1986. *The Control Revolution: Technological and Economic Origins of the Information Society*. Cambridge, MA: Harvard University Press.

Berger, M. C. 1988. "Predicted Future Earnings and Choice of College Major." *Industrial and Labor Relations Review*, vol. 41, pp. 418–29.

Bowen, H. R. and J. H. Schuster, 1986. *American Professors: A National Resource Imperiled*. New York: Oxford University Press.

Bowen, W. G. and N. L. Rudenstine, 1992. *In Pursuit of the PhD*. Princeton, NJ: Princeton University Press.

Braddock, D. 1995. "What Is a Technician?" *Occupational Outlook Quarterly*, vol. 39, no. 1, pp. 38–44.

Bresser, R. K. 1984. "The Context of University Departments: Differences Between Fields of Higher and Lower Levels of Paradigm Development." *Research in Higher Education*, vol. 20, no. 1, pp. 3–15.

Bureau of Labor Statistics. 1994. *Career Guide to Industries*. Washington, DC: author.

Cappelli, P. et al. 1997. *Change at Work*. New York: Oxford University Press.

Clarke, V. A. and Teague, G. J. 1996. "Characterizations of Computing Careers: Students and Professional Disagree." *Computers and Education*, vol. 26, pp. 241–246.

Computing Research Association. 1994. "1992–93 CRA Survey." *Computing Research News*, vol 6., no. 1, pp. 11–15.

Computing Research Association. 1995. "1994 CRA Taulbee Survey." *Computing Research News*, vol. 7, no. 2, pp. 6–9, 16.

Denning, P. J. 1985. "What is Computer Science?" *American Scientist*. vol 73, no. 1, pp. 16–19.

Denning, P. J. et al. 1989. "Computing as a Discipline." *Communications of the Association of Computing Machinery*, vol. 32, no. 1, pp. 9–23.

Dorato, P. and Abdallah, C. 1993. "A Survey of Engineering Education Outside the United States: Implications for the Ideal Engineering Program." *Engineering Education*, vol. 82, no. 4, pp. 212–215.

Educational Testing Service. 1982. *Graduate Record Examinations, 1982–1984: A Description of the Computer Science Test*. Princeton, NJ: Author.

Eerkes, G. L. 1991. "Computer Science Master's Programs." *Communications of the Association of Computing Machinery*, vol. 34, no. 1.

European Federation of National Associations of Engineers (EFNAE). 1975. *Standards for Engineering Qualifications: a Comparative Study in 18 European Countries*. Paris: UNESCO Press.

Fiorito, J. and Dauffenbach, R. C. 1982. "Market and Nonmarket Influences on Curriculum Choice by College Students." *Industrial and Labor Relations Review*, vol. 36, no. 1, pp. 88–103.

Fung, P., O'Shea, T. and Goldson, D. 1993. "Computer Science Students' Perceptions of Learning Formal Reasoning Methods." *International Journal of Mathematical Education in Science and Technology*, vol. 24, pp. 749–759.

Gannon, J. P. 1992a. *Scientists, Engineers, and Technicians in Nonmanufacturing Industries, 1990*. Washington, DC: National Science Foundation. NSF 92–334.

Gannon, J. P. 1992b. *Scientists, Engineers, and Technicians in Manufacturing Industries, 1989*. Washington, DC: National Science Foundation. NSF 92–312.

Gaudin, S. 1996. "Object Programmers Hard to Find, Keep." *Computerworld*, vol. 30, no. 8, p. 14.

Gibbs, N. E. and Tucker, A. B. 1986. "Model Curriculum for a Liberal Arts Degree in Computer Science." *Communications of the Association for Computing Machinery*, vol. 29, no. 3, pp. 202–230.

Geiger, R. 1993. *Research and Relevant Knowledge: American Research Universities Since World War II*. New York: Oxford University Press.

Gokhberg, L. and L. Mindeli, 1995. *Science and Technology in Russia: 1994 Data Book*. Moscow: Center for Science Research and Statistics.

Graduate Record Examinations Board. 1983. *Director of Graduate Programs: 1984 and 1985*. Volume C: Physical Sciences, Mathematics, Engineering. Princeton, NJ: Educational Testing Service.

Graduate Record Examinations Board. 1984. "GRE Candidate Volume, Means, and Standard Deviations, 1964–1983." Unpublished tables prepared for the Study Group Group on the Conditions of Excellence in Higher Education. Princeton, NJ: Author.

Graduate Record Examinations Board. 1986. *Practicing to Take the GRE Computer Science Test*. Princeton, NJ: Educational Testing Service.

Graduate Record Examinations Board. 1993. *Director of Graduate Programs, 14th Edition*. Volume A: Natural Sciences. Princeton, NJ: Educational Testing Service.

Graff, B. I. 1995. "Employment Trends in the Security Brokers and Dealers Industry." *Monthly Labor Review*, vol. 118, no. 9, pp. 20–29.

Grandy, J. 1989. "Models for Developing Computer-Based Indicators of College Student Learning in Computer Science," in C. Adelman (ed.), *Signs and Traces: Model Indicators of College Student Learning in the Disciplines*. Washington, DC: U.S. Department of Education, pp. 11–64.

Grandy, J. 1997. *Science and Engineering Graduates: Career Advancement and Career Change*. Draft Final Report, GRE No.91–14. Princeton, NJ: Educational Testing Service.

Grandy, J. and N. Robertson. 1992. *Change in Field of Study from Undergraduate to Graduate School*. Princeton, NJ: Educational Testing Service. GREB No. 86–12P.

Haber, L. 1996. "Damn, They're Hot! (C++ Programmers)." *Computerworld*, vol. 30, no. 5, p. 101.

Harms, A. A. 1996. "Towards a Foundational Basis of Engineering Classifications." *Engineering Education*, vol 85, no. 4, pp. 303–308.

Howard, A. 1986. "College Experiences and Managerial Performance." *Journal of Applied Psychology*, vol. 71, pp. 530–552.

Huckenpohler, J. G. 1995. *Selected Data on Graduate Students and Postdoctorates in Science and Engineering: Fall, 1993*. Arlington, VA: National Science Foundation.

Huckenpohler, J. G. 1992. *Selected Data on Graduate Students and Postdoctorates in Science and Engineering: Fall, 1991*. Washington, DC: National Science Foundation.

Kutscher, R. E. 1992. "New BLS Projections: Findings and Implications," in Bureau of Labor Statistics, *Outlook, 1990–2005*. Washington, DC: Author, pp. 1–28.

Lane, M. J. 1988. *Profiles—Computer Sciences: Human Resources and Funding*. Washington, DC: National Science Foundation. NSF 88–324.

Loui, M. C. 1987. "Computer Science is an Engineering Discipline." *Engineering Education*, vol. 78, no. 3, pp. 175–179.

Machlup, F. 1980. *Knowledge: Its Creation, Distribution and Economic Significance*. Princeton, NJ: Princeton University Press.

Morrison, R. E. 1995. *Scientists, Engineers, and Technicians in Manufacturing Industries, 1992*. Arlington, VA: National Science Foundation.

Mueller, M. 1989. "Yellow Stripes and Dead Armadillos," in P. Franklin (ed.), *Profession, 1989*. New York, NY: the Modern Language Association.

Mulder, M. C. and Dalphin, J. 1984. "Computer Science Program Requirements and Accreditation—an Interim Report of the ACM/IEEE Computer Society Joint Task Force." *Communications of the Association for Computing Machinery*, vol. 27, no. 4, pp. 330–335.

National Research Council. 1993. *Computing Professionals: Changing Needs for the 1990s*. Washington, DC: Author.

Organization for Economic Cooperation and Development. 1992. *From Higher Education to Employment*. 3 vols. Paris: Author.

Oltman, P. K. 1982. *Content Representativeness of the GRE Advanced Tests in Chemistry, Computer Science, and Education*. GREB 81-12p. Princeton, NJ: Educational Testing Service.

Oltman, P. K. and R. T. Hartnett. 1984. *The Role of GRE General and Subject Test Scores in Graduate Program Admission*. GREB No. 81-8R. Princeton, NJ: Educational Testing Service.

Owens, W. A. 1976. "Background Data," in M. Dunnette, ed. *Handbook of Industrial Psychology*. Chicago: Rand-McNally, pp. 609–644.

Pacey, A. 1983. *The Culture of Technology*. Cambridge, MA: MIT Press.

Palafox, J., J. G. Mora, and F. Pérez. 1995. *Capital Humano, Educacion y Empleo*. Valencia: Fundació Bancaja.

Pantin, C. F. A. 1968. *The Relations Between the Sciences*. Cambridge (England): Cambridge University Press.

Polachek, S. W. 1978. "Sex Differences in College Major." *Industrial and Labor Relations Review*, vol. 31, pp. 498–508.

Powers, D.E. and M. K. Enright. 1986. *Analytical Reasoning Skills Involved in Graduate Study: Perceptions of Faculty in Six Fields*. GREB #83-23p. Princeton, NJ: Educational Testing Service.

Rosenthal, N. H. 1992. "How Accurate Are the Employment Projections?" *Occupational Outlook Quarterly*, vol. 36, no. 3, pp. 24–32.

Silvestri, G. and Lukasiewicz, J. 1992. "Occupational Employment Projections," in Bureau of Labor Statistics, *Outlook, 1990–2005*. Washington, DC: Author, pp. 62–121.

Simsek, H. and Louis, K. S. 1994. "Organizational Change as Paradigm Shift: Analysis of the Change Process in a Large, Public University." *The Journal of Higher Education*, vol. 65, no. 6, pp. 670–695.

Smith, B. L. R. (ed). 1985. *The State of Graduate Education*. Washington, DC: the Brookings Institution.

Smith, T. M. et al. 1996. *The Condition of Education, 1996*. Washington, DC: National Center for Education Statistics.

Snyder, T. 1995. *Digest of Education Statistics, 1995*. Washington, DC: National Center for Education Statistics.

Snyder, T., C. M. Hoffman, and C. M. Geddes. 1996. *Digest of Education Statistics, 1996*. Washington, DC: National Center for Education Statistics.

Tsapogas, J. 1992. *Characteristics of Recent Science and Engineering Graduates, 1990*. Washington, DC: National Science Foundation.

Tucker, A. B. et al. 1991. *Computing Curricula 1991: Report of the ACM/IEEE-CS Joint Curriculum Task Force*. New York: Association for Computing Machinery.

Turner, A. J. 1993. "U.S. Degree Programs in Computing." In *Computing Professionals: Changing Needs for the 1990s*. Washington, DC: National Research Council, pp. 120–135.

U.S. Bureau of the Census. 1992. *Current Population Survey* (March). Washington, DC: Author.

Wilde, C. 1996. "Steaming Brew (Demand for and Competition Among Java Programmers." *Computerworld*, vol. 30, no. 6, p. 87.

Wilkinson, R. K. 1990. *Science and Engineering Personnel: a National Overview*. Washington, DC: National Science Foundation.

Wulf, W. A. 1992. "Computer Science and Software Engineering: Splitting is the Wrong Solution." *Computer Science Education*, vol. 3, no. 2, pp. 123–134.

Appendix: Technical Notes and Guidance

There are many tables in this document, both in the text and in the notes. Some are derived or constructed from other published sources. Where those sources are a complete census, for example, a survey of all computer science departments conducted by the Mathematical Association of America, we don't worry much about samples and weights. Where those sources rely on samples, we must assume that the statistical standards of agencies such as the National Science Foundation, the Bureau of Labor Statistics, and the Bureau of the Census were observed in the production of data.

But 15 of the tables were prepared using special analyses files created from two longitudinal studies of the National Center for Education Statistics: the National Longitudinal Study of the High School Class of 1972 (NLS-72) and the High School and Beyond/Sophomore Cohort (HS&B/So), and it is helpful to know something about the statistical standards that lie behind these tables and the decision rules that were used in presenting the data.

The populations in these longitudinal studies are national probability samples first drawn when the students were in high school. They involve first, a stratified sample of secondary schools with an over-sampling of schools in minority areas, and a random sampling of students at a specific grade level (12th grade for the NLS-72, 10th grade for the HS&B/So) within those schools. The original sample is then weighted to match the national census, for example, of all 10th-graders in 1980 (about 3.7 million people). Each participant carries a weight in inverse proportion to the probability that he or she would be selected by chance. In both longitudinal studies, the original samples were what statisticians call "robust": 22,650 for the NLS-72 and 28,000 for the HS&B/So. After the base year of these surveys, every subsequent survey is a subset of the original, and the weights carried by participants are modified accordingly. In the penultimate survey of the NLS-72 in 1986, there were 12,841 respondents out of 14,489 surveyed; in that for the HS&B/So in 1992, there were 12,640 respondents out of 14,825. The postsecondary transcript file for the NLS-72 has 12,599 cases; that for the HS&B/So has 8,395 cases. These are still very robust numbers. They represent populations in the millions. By the conclusion of any of these longitudinal studies, a student is carrying a half-dozen different weights, depending on what question is asked.

For the High School and Beyond cohort, for example, I used three different weights in the tables in this study: a "senior year" weight for a question such as the relationship between the highest level of mathematics studied in high school and whether someone eventually earns a bachelor's degree; a "postsecondary transcript weight" for analyses of what students study in college; and a "final weight" for 1991 labor market experience analyses.

More important are issues of standard errors of measurement and significance testing. What you see in the tables are estimates derived from samples. Two kinds of errors occur when samples are at issue: errors in sampling itself, particularly when relatively small subpopulations are involved, and non-sampling errors. Non-sampling errors are serious matters. Good examples would include non-response to specific questions in a survey or

missing college transcripts. Weighing will not address the panoply of sources of non-sampling errors.

The effects of sampling and non-sampling errors ripple through data bases, and, to judge the accuracy of any analysis, one needs to know those effects. When the unit of analysis is the student, this is a straightforward issue. When we ask questions about bachelor's degree majors (table 4), number of completed courses in computer science (table 5), or 1991 earnings (table 17), we are asking questions about non-repetitive behaviors of people who were sampled. To judge comparisons in these cases we use the classic "Student's *t*" statistic that requires standard errors of the mean. But because the longitudinal studies were not based on simple random samples of students, the technique for generating standard errors involves a more complex approach known as the Taylor series method. For the descriptive statistics in this report, a proprietary program incorporating the Taylor series method, called STRATTAB, was used.

It is important to note that STRATTAB will provide neither estimates nor standard errors for any cell in a table in which the unweighted N is less than 30. For those cells, the program shows "LOW N." Table 5 on page 14 illustrates the frequency of LOW N cells that occur when one is making multiple comparisons among categories of an independent variable.

Table 5 also illustrates a basic decision rule followed in the presentation of data: with only one exception, I displayed neither standard errors of measurement ("s.e.") nor indications of statistical significance on the basis of two-tailed tests of "Student's *t*" in complex tables with many cells (for example, tables 5, 6, and 7). In the case of s.e.'s, there is simply too much already on the page. In the case of indicating statistical significance of pairs of cells, it would only be distracting the reader's attention from the point of the table. In cases where the two longitudinal studies cohorts are compared, too, the standard errors refer to within-cohort estimates, not time series estimates. For presentations such as tables 6 and 7, the reader can be assured that the within-cohort standard errors yield common sense judgments of statistical significance, for example, that there is no difference between a course claiming 1.1 percent of time and another course claiming 1.3 percent of time.

In the case of simple estimates for which a reader may be interested in comparing two categories of a dependent variable, standard errors are offered, and the reader can employ the basic formula for computing the "Student's *t*":

$$t=(P_1 - P_2) \sqrt{(se_1^2 + se_2^2)}$$

where P_1 and P_2 are the estimates to be compared and se_1 and se_2 are the corresponding standard errors. If, in this case, $t \geq 1.96$, you have a statistically significant difference such that the probability that this observation would occur by chance is less than 1:20. In the case of multiple comparisons, the critical value for t rises following the formula for Bonferroni Tests: if H comparisons are possible, the critical value for a two-sided test is $Z_{(1-.05/2H)}$.

★ U.S. GOVERNMENT PRINTING OFFICE: 1997 – 426 - 843 / 70331

Standard Mail (A)

PLLI 97, 8046